

**Redakční systém vědeckého
časopisu založený na
typografickém prostředí L^AT_EX**

**Content management system of
scientific journal, based on the
typographical program L^AT_EX**

Zadání diplomové práce

Student: **Bc. Jakub Fidler**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T059 Mobilní technologie

Téma: **Redakční systém vědeckého časopisu založený na typografickém prostředí LaTeX**
Content Management System of Scientific Journal, Based on the Typographical Program LaTeX

Zásady pro vypracování:

1. Navrhněte systém pro sazbu vědeckého časopisu na bázi typografického prostředí LaTeX s možností exportu metadat a automatickou tvorbou jednotlivých čísel časopisu pomocí skriptů.
2. Pro export metadat sloužících např. k indexaci v databázích, nebo pro potřeby digitálních knihovnických systémů použijte prostředí Tralics.
3. Navržený systém otestujte na serveru se systémem Linux a některém open source softwarem pro správu a publikování časopisů.
4. Navržený systém důkladně zdokumentujte a vytvořte obraz celého navrženého systému včetně operačního systému a všech potřebných doplňků.
5. Jako příloha diplomové práce budou i open source nástroje pro spuštění vytvořeného obrazu, pro platformy Linux i MS Windows.

Pro vypracování závěrečné práce bude použit typografický systém LaTeX.

Seznam doporučené odborné literatury:

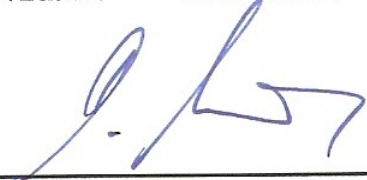
Oetiker, T., a kol. *Ne příliš stručný úvod do systému LaTeX 2e*.

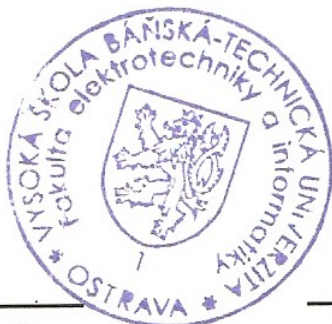
Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí diplomové práce: **Ing. Jan Skapa, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry




prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 24. dubna 2014


.....

Chtěl bych velmi poděkovat svému vedoucímu práce Ing. Janu Skapovi, Ph.D. , za jeho čas strávený u konzultací a poskytnutí důležitých rad, které mi pomohly při tvorbě diplomové práce. Dále děkuji své rodině za vytvoření ideálních studijních podmínek nejen při psaní diplomové práce, ale také po celou dobu studia na vysoké škole.

Abstrakt

Cílem této diplomové práce je navrhnout systém pro sazbu vědeckého časopisu Advances in Electrical and Electronic Engineering v prostředí \LaTeX s možností exportu metadat a automatickou tvorbou jednotlivých vydání. Daný systém je po vytvoření otestován na open source softwaru, který je určen pro správu a publikování časopisů. Vytvořený systém pracuje na základě bash skriptů postupně překládajících jednotlivé stránky napsané v \LaTeX u, využívajících informací uložených v souboru s příponou csv. Za pomoci programu Tralisc generují metadata článků ve formátu XML. Pro otestování návaznosti na volném redakčním systému byl použit redakční systém OJS. Vytvořený systém dokáže poskládat zvolené vydání časopisu do jednoho dokumentu, vytvořit verzi pro online publikování s vygenerovanými metadaty a verzi pro tisk. Přínosem této práce je zrychlení a ulehčení tvorby jednotlivých vydání vědeckého časopisu.

Klíčová slova: typografie, \LaTeX , XML, bash, csv, časopis, automatický systém , OJS

Abstract

The goal of this thesis is to design a system for typesetting scientific journal Advances in Electrical and Electronic Engineering in \LaTeX environment with the ability to export metadata and to create each issue automatically. The created system is tested on open source software which is designed to manage and publish magazines. The created system works on the basis of bash scripts sequentially translating individual pages written in \LaTeX , using information stored in a csv file. With Tralisc they generate metadata of the articles in the XML format. CMS OJS was used to test a relation to open free content management system. The created system can put together selected issue in a single document, create a version for online publishing with generated metadata and a printable version.

Keywords: typography, \LaTeX , XML, bash, csv, jurnal, automatic system, OJS

Seznam použitých zkratek a symbolů

XML	– Extensible Markup Language
CSV	– Comma-separated values
DTD	– Document Type Definition
PDF	– Portable Document Format
OJS	– Open Jurnal System
CSM	– Content Management System
GPL	– General Public License

Obsah

1	Úvod	4
2	Vědecký časopis AEEE	6
2.1	Zaměření časopisu	6
2.2	Frekvence vydání	6
2.3	Vydání z konferencí	7
2.4	Proces publikování článku v AEEE	7
3	T_EX	8
3.1	Vznik a historie T _E Xu	8
3.2	Způsob tvorby stránky	8
3.3	Cyklus zpracování textu T _E Xem	10
3.4	Typografický systém L ^A T _E X	11
4	Bash	14
4.1	První skript	14
4.2	Vybrané bash příkazy	16
5	Tralics	18
5.1	Struktura L ^A T _E Xového dokumentu	18
5.2	Konfigurační soubory	19
5.3	Přepínače pro Tralics	20
6	Open Jurnal System	22
6.1	Funkce OJS	22
6.2	Instalace OJS	23
6.3	Role OJS	23
6.4	Přídavné moduly	26
7	Systém pro automatickou tvorbu časopisu	27
7.1	Ruční tvorba časopisu	27
7.2	Možnosti řešení	27
7.3	Schéma systému	27
7.4	Adresářová struktura systému	28
7.5	Popis jednotlivých částí systému	29
7.6	Skripty	37
8	Testování vytvořeného systému	44
8.1	První použití vytvořeného systému	44
8.2	Testování s OJS	44
8.3	Zhodnocení testování	45
9	Závěr	46

10 Reference	47
Přílohy	48
A Vývojové diagramy bash skriptů	49
B Postup zprovoznění systému na OS Linux	54
C Obsah DVD	55

Seznam obrázků

1	Ukázka webových stránek časopisu AEEE	7
2	Písmová osnova	9
3	Graf používání OJS ve světě [17]	22
4	Popis procesů probíhajících při tvorbě časopisu	24
5	Schéma systému	29
6	Souborová struktura systému	30
7	Ukázka úvodního slova daného vydání	31
8	Ukázka obsahu daného vydání	32
9	Adresářová struktura článku	34
10	Formátování textu PDF/TXT	35
11	Titulní stránka časopisu AEEE	35
12	SOURCE.CSV	36
13	Volba vydání	37
14	Proces nahrazování znaků	38
15	Překlad článků	40
16	Vývojový diagram skriptu spoj.sh	42
17	List s ořezovými značkami	43
18	Ukázka uživatelů v OJS	45
19	Vývojový diagram skriptu start.sh	49
20	Vývojový diagram skriptu about.sh	50
21	Vývojový diagram skriptu clear.sh	51
22	Vývojový diagram skriptu uvod_preklad.sh	52
23	Vývojový diagram skriptu clanek_preklad.sh	53

1 Úvod

Na VŠB-TU se vydává několik různých vědeckých časopisů, mezi které patří i časopis *Advances in Electrical and Electronic Engineering*. Jmenovaný časopis je vydáván katedrou telekomunikační techniky. V předešlých letech redakce časopisu přijímala od autorů články v různých datových formátech. V tomto seznamu se nacházely články psané pomocí typografického systému \LaTeX . Tento systém pro profesionální sazbu textu byl zvolen vydavatelem jako výchozí formát pro přijímání vědeckých článků. Volba jednotného formátu, umožnila využít myšlenku zautomatizování procesu tvorby celého časopisu. Což by mělo za důsledek zkrácení doby na přípravu a především eliminaci tvorby chyb a překlepů. Redakce přijímá články, které procházejí běžnými procesy při publikaci časopisu, až se dostanou do stavu připraveného k vydání. V této fázi se všem stránkám nejen u článků, ale i u zbytku stránek časopisu, musí přidat do záhlaví a zápatí informace o daném vydání. Dále se musí vytvořit úvod, úvodní slovo a v neposlední řadě složit všechny články do jednoho celistvého dokumentu pro online publikaci a pro verzi k tiskařskému vydání. Což je pro manuální činnost velice náročné.

V první části se má diplomová práce zabývat teoretickým popisem. Úvodní kapitola popisuje samotný časopis, pro který je tento systém tvořen, je zde popsána historie a zaměření časopisu, dále redakční procesy, pomocí kterých celý časopis vzniká. V následující části má diplomová práce popisovat zvolené technologie a programy, které vstupují do tvorby vytvářeného systému pro automatickou publikaci.

Všechny publikované dokumenty jsou psány pomocí technologie \LaTeX , ve kterém jsou tvořeny jednotlivé články i titulní strany časopisu. Je zde popsán vznik, popis zpracování textu pomocí \LaTeX u, výhody a nevýhody, ukázka vytvoření minimálního \LaTeX ového souboru, či tvorba vlastních \LaTeX ových příkazů.

Aby bylo možné celý proces skládání vydání časopisu řídit, bylo potřeba využít bashovských skriptů, jimž se věnuje další teoretická část mé diplomové práce. V této části jsou popsány základní prvky bash skriptování, zakončené výčtem nejpoužívanějších příkazů v vytvořeném automatizovaném systému.

Další kapitola se zabývá vytvářením XML souborů z \LaTeX ových zdrojů. Tato funkce slouží při exportu metadat pro jednotlivé články, jenž jsou použity pro indexaci v digitálních knihovnách. Tvorba dokumentů je prováděna pomocí programu Tralics, který umožňuje převádět \LaTeX ové soubory do formátu XML. V kapitole je popsán obecný popis tohoto programu, ukázány základní struktury potřebného dokumentu pro převod do XML, nastavení konfiguračních souborů, popřípadě výčet přepínačů pro usměrnění výstupu tohoto programu.

Pro publikaci časopisu v určité kvalitě, je zapotřebí, aby články prošly velkým množstvím procesů a kontrol. K vytvoření kvalitního dokumentu dopomáhají softwary jako například redakční systémy. Některé CSM jsou přímo uzpůsobeny pro vydávání časopisů. Mezi takové redakční systémy patří například software Open Jurnal Systém (dále jen OJS). V této sekci jsou popsány obecné informace, způsob instalace, jeho funkce a role které poskytuje pro jednotlivé uživatele.

Druhá část mé diplomové práce se týká vytvořeného systému pro automatickou tvorbu časopisu, který je schopen poskládat celé vydání do jednoho konečného dokumentu připraveného ve verzích pro tisk a online publikování. Celý tento proces je řízen pomocí bash skriptů, jenž čerpají informace z csv soboru, kde se definují parametry vytvářeného vydání. Celý systém umí automaticky ze zadaných informací vytvořit obsah, úvodní proslov, přidat záhlaví a zápatí nejen samotným článkům, ale také ostatním listům časopisu. Případně přidat ořezové značky pro tiskovou verzi. Pro online verzi je systém schopen vygenerovat ke každému článku metadata ve formátu XML.

Samotné kapitoly zabývají popisem jednotlivých částí vytvořeného systému a poté postupem, kde jsou detailně popsány jednotlivé kroky. Samotné bash skripty krom textového popisu jsou znázorněny na vývojových diagramech přiložených v příloze této práce.

Závěrečná část práce obsahuje testování vytvořeného systému s Open Jurnal System na operačním systému Linux. Jsou zde popsány postupy pro zprovoznění systému na čisté instalaci. Popis fáze publikování, v které se vytvořený systém používá. A určení osoby pro její obsluhu. A celkové zhodnocení testování.

Práce také obsahuje přílohy, ve kterých je popsán postup zprovoznění systému na čisté instalaci OS Linux, obsah příkládaného DVD a vývojové diagramy bash skriptů.

2 Vědecký časopis AEEE

Vědecký časopis *Advances in Electrical and Electronic Engineering* (AEEE) byl vydáván v tištěné podobě na Fakultě elektrotechniky, University of Žilina v Žilině, mezi léty 2002 až 2008. V roce 2009 bylo rozhodnuto o přesunutí časopisu AEEE na VŠB-TU Ostrava. Vzhledem ke složitému přenosu časopisu, bylo jeho vydávání v roce 2009 přerušeno. Od roku 2010 se AEEE vydává v elektronické podobě na VŠB - Technické univerzitě Ostrava společně s Fakultou elektrotechniky, University of Žilina, v rámci dohody o partnerství s ISSN 1804 - 3119. Příspěvky publikované v časopise 2002-2008 jsou uloženy v archivu časopisu. Od roku 2011 je vědecký časopis AEEE publikován také v tištěné podobě pod ISSN 1336-1376. Vědecký časopis AEEE je indexován v hlavních světových databázích, jako jsou DOAJ, Driver, Google Scholar a EBSCO. Od roku 2011 je časopis *Advances in Electrical and Electronic Engineering* také indexován a publikován v jedné z nejdůležitějších a nejvlivnějších databází na světě SciVerse Scopus¹. Časopis AEEE je také partnerem významných mezinárodních konferencí, z nichž jsou publikovány nedůležitější příspěvky vybrané mezinárodním vědeckým výborem dané konference. [4]

2.1 Zaměření časopisu

AEEE je vědecký časopis zaměřený na publikování výsledků výzkumu z vědeckých oblastí. Cílem editorů je publikovat kvalitní vědecké odborné dokumenty, které mohou být prezentovány od významných vědeckých týmů a zkušených autorů, až po postgraduální studenty a začínající výzkumné pracovníky. Jedním z hlavních úkolů časopisu je usnadnit kontakty mezi výzkumnými středisky a průmyslem. Všechny články jsou vystaveny anonymnímu procesu přezkoumání dvěma odbornými recenzenty před zveřejněním na webových stránkách AEEE².

2.2 Frekvence vydání

Články v tomto časopise jsou publikovány nakladatelem čtvrtletně. Na jaře (březen), v létě (červen), na podzim (září) a zimě (prosinec). Časopis *Advances in Electrical and Electronic Engineering* se kromě klasického čtvrtletního vydání publikuje i v elektronické on-line verzi, vždy:

- 31. března
- 30. června
- 30. září
- 31. prosince

¹<http://www.scopus.com/home.url>

²<http://advances.utc.sk/index.php/AEEE/index>



Obrázek 1: Ukázka webových stránek časopisu AEEE

Články přijímá redakce časopisu po celý rok. Doba mezi přijmutím a publikováním odborného článku se pohybuje mezi třemi až šesti měsíci [5]. Na obrázku č. 1, je ukázka webu časopisu AEEE.

2.3 Vydání z konferencí

Časopis nezveřejňuje konferenční materiály (např. dokumenty použité na konferenci). Výjimku můžou tvořit pozvání účastníci, hlavní řečník nebo vybrané práce. Nicméně, je potřeba prezentované dokumenty z konference rozšířit, aby mohly být publikovány. Články podané do časopisu by se měly lišit od dříve publikovaných materiálů minimálně o 40%. Redakce poté vyžaduje zaslání sborníku (na CD) pro vyhodnocení rozšíření článku poslaného ke zveřejnění ve zvláštním vydání. Rozhodnutí o přijetí / odmítnutí je založeno na doporučení dvou nezávislých recenzentů. Jeden z nich musí být členem ve vědeckém výboru konference a druhý v redakci časopisu.

Redakční rada může odmítnout další dokumenty na základě doporučení dalších posuzovatelů ve sporných případech. Příspěvky z dané konference budou publikovány společně ve zvláštním vydání časopisu AEEE [5].

2.4 Proces publikování článku v AEEE

Cílem tohoto komplikovaného procesu je zajistit vysokou kvalitu odborného časopisu. Recenzent má za úkol odborné posouzení článku přijatého od autora v elektronické podobě, podle definované šablony. Recenzent připomínky a doporučení shrne do formuláře pro hodnocení podaných článků. Autoři by poté měli dané nejasnosti vyhodnotit a opravit před zveřejněním. Články, které recenzenti nedoporučí k publikování nejsou zveřejněny.

Po doporučení recenzentem, je každý článek kontrolován dalšími osobami (section editor, layout editor a korektor).

Doba pro zpracování odborných posudků je obvykle asi 8 týdnů. Během tohoto období recenzenti zpracovávají vzájemné hodnocení a vyjadřují své názory [5].

3 T_EX

Kapitola obsahuje základní informace o způsobu sazby textu pomocí T_EXu. Popisuje jeho historii a způsob vytváření dokumentů pomocí této technologie. Dále se zaměří na jeho rozšíření L^AT_EX. Kde budou popsány základní příkazy pro tvorbu v tomto typografickém systému.

3.1 Vznik a historie T_EXu

Když byl v roce 1977 profesor Donald E. Knuthen ze Stanfordské univerzity nespokojen se sazbou své knihy „*The Art of Computer Programming*“, rozhodl se vytvořit počítačový program T_EX. Následně se Knuthen seznámil s možnostmi digitální fotosazby a vytvořil speciální jazyk počítačové typografie T_EX. První bližší seznámení s tímto programem provedl na přednášce pro Americkou matematickou společnost, kde vyzdvihl jeho přednosti. Šlo především o kvalitu sazby matematických textů. V roce 1978 se začala používat první verze T_EXu označovaná jako T_EX78. Druhou verzí je T_EX82, která obsahuje vybavenější skupinu fontů Computer Modern. Knuthen. Tuto verzi poté přepracoval a vznikla verze T_EX3.0. Největší změnou u verze 3.0 byla možnost pracovat se vstupním souborem, který obsahoval znaky s kódem 0-255 (v předchozích verzích bylo podporováno pouze 128 kódů). Navíc byla tato verze již multilinguální, to znamenalo, že v jednom textu bylo možno dělit slova podle pravidel různých jazyků.[1, 3]

Největší přemostí T_EXu je vynikající způsob formátování matematických zápisů. Pro začínající uživatele je však tento systém celkem složitý. Kvůli nepřátelsky působícímu programu, byly vytvořeny soubory maker, které měly usnadnit zpracování sázeného textu. Mezi nejznámější patří PlainT_EX, L^AT_EX, AMSL^AT_EXa LamST_EX. [1]

PlainT_EX- nejjednodušší nadstavba, která se používá jako základ ostatních maker.

L^AT_EX - užívá se především pro tvorbu rozsáhlých publikací, a to nejen matematických, podporuje tvorbu obsahu, rejstříku a křížových referencí.

AMSL^AT_EX - hlavním účelem makra je tvořit kratší složitě matematické sazby.

LamST_EX - umožňuje rozšiřovat makra o nové další funkce. Dálo by se říct, že je to kombinace AMSL^AT_EXu a L^AT_EXu.

3.2 Způsob tvorby stránky

Tvorba stránky je složena ze dvou fází, které probíhají skoro paralelně. První fáze čte vstupní soubor a každý znak dokumentu přiřadí do určité kategorie. Zároveň ze vstupního textu jsou vypuštěny nadbytečné mezery a komentáře.

Druhou fází je vlastní sazba stránky. Obecně lze říci, že se postupuje od nejjednodušších objektů k nejsložitějším. Za nejjednodušší objekt považujeme písmeno, které je charakterizováno svou šířkou, výškou nad účaří a hloubkou pod účaří. (Obrázek č. 2). Tato písmena jsou pak na základě těchto údajů skládány do slov. Slova s pomocí mezer



Obrázek 2: Písmová osnova

(mezery kromě své optimální velikosti mají definovanou velikost o kolik se mohou zúžit popřípadě roztáhnout) tvoří jednotlivé řádky. Ty jsou pak následně sloučeny do odstavců. A odstavce po složení již dají finální objekt (stránku).

Předchozí popis je velice zjednodušen a nepočítá se množstvím možností, jako je například s umístěním tabulek na stránku, obtékáním textu kolem obrázků, sazbu do sloupců, horním a dolním záhlavím, poznámkami pod čarou a mnoha jinými.

Celá sazba pomocí \TeX je velice komplikovaná, dle potřeby je možné se automaticky pohybovat mezi 6 režimy práce. Každý režim má na starost své speciální úkoly. Například skládání písmenek na řádku a do slov, skládání řádek do odstavců, skládání odstavců na stránku (i do více sloupců), doplňování stránky o záhlaví, sazbu matematických vzorců a podobně.

Příhodná je ukázka tvorby odstavce. \TeX se prvně pokusí vysázet odstavec tak, aby nemusel použít dělení slov. Toto je velice rychlá a nenáročná operace, protože nemusí zjišťovat, jak je které místo slova vhodné k dělení. Pokud se mu to nepodaří, respektive šířka řádky je krátká, doplní jednotlivá slova o informace o vhodnosti dělení a poté celý odstavec znovu zpracuje, tentokrát s možností dělení slov. \TeX má i další velmi užitečné funkce, jenž například dokážou kontrolovat, zda odstavec nezačíná na posledním řádku stránky, nebo naopak aby nekončil na prvním řádku stránky nové. Popřípadě kontroluje, aby poslední slovo v odstavci nebylo na samostatné řádce. \TeX obsahuje mnoho dalších podobných šikovných funkcí, jenž ve výsledku nebývale pozvednou celkový vzhled a estetický dojem dokumentu [1].

3.3 Cyklus zpracování textu \TeX em

Celá tvorba dokumentu za pomoci \TeX u probíhá v několika cyklech. Přesněji ve třech. První cyklus se stará o přípravu vstupního textu za pomoci libovolného textového editoru. Tento vstupní soubor obsahuje kromě samotného textu i další informace. Jsou to řídicí posloupnosti, jež udávají vzhled dokumentu. Respektive určují formát stránky, odstavce, řádku a mnoho dalších. Samotný \TeX obsahuje až 300 jednoduchých funkcí a další složitější posloupnosti mohou být určovány balíky maker v průběhu zpracování textu (například balík maker \LaTeX).

Primitivní funkce se rozdělují podle svých činností do několika skupin:

- primitivní funkce řídící zlom a zarovnávání odstavců, a to včetně prostředků pro ovlivnění dělení slov;
- primitivní funkce ovlivňující celkový formát stránky (velikost a umístění, zlom stránky, horní a dolní záhlaví, poznámky pod čarou, umístění obrázků a tabulek);
- primitivní funkce řídící velikosti mezer (mezi písmeny, slovy, odstavci a symboly v matematických vzorcích);
- primitivní funkce pro speciální zarovnávání textu (tabulky, matice atd.);
- primitivní funkce umožňující definování nových řídicích posloupností;
- primitivní funkce měnící chování \TeX u;
- primitivní funkce ovládající práci \TeX u;
- primitivní funkce umožňující pracovat s externími soubory (začleňovat je do vstupního souboru, měnit je a vytvářet nové);

Poté se již pracuje s připraveným vstupním souborem z prvního cyklu, která nechá dokument přeložit \TeX em, jenž vytvoří soubor s příponou `.dvi`³. Tento soubor je tvořen stránkami, jež obsahují kolekce kódů a příslušných fontů s informacemi o jejich umístění na dané stránce. Tyto informace se poté využijí pro vytištění popřípadě zobrazení na jakémkoliv výstupním zařízení. Což patří mezi velké výhody \TeX u. V samotném důsledku program umožňuje tisknout dokumenty na „čemkoliv“. Samotné zobrazení dokumentu, přesněji *DVI* souboru, řídí ovladač příslušného výstupního zařízení, který z údajů *dvi* souboru a bitových map jednotlivých písmenek složí finální podobu dokumentu [1].

Ve většině běžných textových aplikací se pro tisk matematických symbolů popřípadě řeckých písmen a podobných znaků využívají různé klávesnice, \TeX má v sobě zabudované makra pro jejich sazbu. Pokud tedy chceme například napsat ω , stačí do vstupního souboru zapsat `ω`. Jednotlivé řídicí zkratky jsou většinou odvozeny od jejich anglických názvů, především proto, že \TeX vznikl v Spojených státech Amerických [1].

³soubor (device independant)

3.4 Typografický systém \LaTeX

Jak již bylo zmíněno \LaTeX je balík maker, jenž umožňují autorům sázet a tisknout dokumenty v nejvyšší možné typografické kvalitě. \LaTeX byl původně vytvořen Leslie Lamportem. A používá \TeX pro sazbu textu. V současné době je \LaTeX rozšiřován týmem $\text{\LaTeX}3$. Hlavním úkolem tohoto týmu je sjednotit jednotlivé verze, a dále je rozšiřovat [2].

3.4.1 Výhody a nevýhody systému \LaTeX

Aby bylo možné uvést výhody a nevýhody je potřeba \LaTeX s něčím porovnat. Nejlepší příklad bude \LaTeX postavit proti WYSIWYG⁴ aplikacím. Toto jsou aplikace, u kterých uživatel okamžitě vidí vzhled stránky a formátování stránky se řídí interaktivně. Mezi tyto systémy můžeme uvést Word for Windows, Tex602 a mnoho dalších [2].

Mezi největší výhody \LaTeX u bezpochyby patří tvorba rozsáhlých děl se složitou strukturou, jenž obsahuje krom samotného textu, obrázky tabulky seznamy použité literatury až po tištění matematických vzorců, jenž bývají uváděny jako jedny z dalších velkých výhod. Pokud potřebujeme použít nějaké speciální typografické funkce, existují různé nadstavby v podobě balíčků maker. Například přesně standardizované vytváření bibliografických záznamů. Všechny tyto funkce dohromady dávají systém, jenž ve svém výsledku vytvoří profesionálně formátovaný dokument. V neposlední řadě je potřeba také zmínit, že \LaTeX můžeme využívat na všech druzích operačních systémů. Dokonce i na operační systému Android⁵

Mezi nevýhody \LaTeX u můžeme zařadit jeho instalaci. Kompletní instalace zabírá celkem velké množství diskového prostoru. Dále lze vzít v potaz pracnější vytváření formátu stránky, pokud nemáme předem definované styly.

3.4.2 Soubor s příponou \TeX

Jako vstupní soubor se do \LaTeX u používá textový ASCII soubor, vytvořený libovolným textovým editorem. Toto umožňuje vytvářet \TeX ové soubory pomocí práce s textem, jenž je uvedena v pozdějších kapitole 4. Důležité je podotknout, že \LaTeX ignoruje více mezer za sebou, tabulátor považuje za mezeru. Jako ukončení odstavce bere prázdný řádek. Pokud kód obsahuje více prázdných řádků je to opět latexem brán jako jeden řádek. Kromě těchto vychytávek má \LaTeX předdefinované speciální znaky, které nejsou vysázeny jako klasický text, ale mají specifickou funkci. Mezi tyto znaky patří $\&$ $\$$ $\%$ $\#$ $_$ $\{$ $\}$. Aby bylo možné tyto znaky vysázet jako klasický text, musí se přidat před daný znak \backslash Anglicky název je *backslash*. Zápis poté vypadá $\backslash\&$ $\backslash\$$ $\backslash\%$ $\backslash\#$ $\backslash_$ $\backslash\{$ $\backslash\}$. [2]

3.4.3 Struktura vstupního souboru

Kromě samotného textu obsahuje vstupní soubor příkazy, jenž definují chování dokumentu. Mezi nepostradatelné příkazy, které tvoří základní strukturu dokumentu, patří

⁴“What you see is what you get”, způsob tvoření dokumentů. To co je zobrazené je i výsledná podoba dokumentu)

⁵<https://code.google.com/p/texlive-for-android/>

`\documentclass {}`. Tento příkaz specifikuje, jakého druhu bude daný dokument. Popřípadě zde mohou být uvedeny informace, jež ovlivňují vzhled budoucího dokumentu. Dalším příkazem je `\usepackage`, pomocí kterého načítáme balíky dalších funkcí s novými příkazy. Samotné tělo dokumentu poté ohraničují dva příkazy `\begin{document}` a `\end{document}`. Mezi touto dvojicí se nachází vlastní text dokumentu smíchaný s dalšími příkazy \LaTeX u. Vše co se nachází za příkazem `\end{document}` bude ignorováno. Na výpisu je ukázán minimální soubor \LaTeX u. Tento soubor obsahuje výše uvedené příkazy. [2]

```
\documentclass{article}
\begin{document}
    Dokument je na svete!
\end{document}
```

Jak již bylo zmíněno prvním příkazem v dokumentu je `\documentclass`, který dává \LaTeX u informace o typu dokumentu, který chce vytvářet. Zda se jedná o třídu `article` používající se především pro psaní odborných článků, krátké zprávy prezentace a podobně nebo `report`, který se používá především při vytváření malých knih diplomových prací či delších zpráv. Popřípadě třída `book` používající se pro sazbu skutečných knih.

3.4.4 Výběr \LaTeX ových příkazů

Příkazy, jež se používají v \LaTeX u jsou citlivá na velká a malá písmena. Jejich formát je následující:

- Začínají znakem `\` pokračujícím názvem příkazu, který je složený pouze z písmen. Například `\today`.
- Některé příkazy vyžadují vstupní parametr, který se udává do složených závorek. Například `\emph{zvyrazнено}`
- Určité příkazy dovolují zadávat i nepovinné parametry v hranatých závorkách. Například `\linebreak[2]`

\LaTeX obsahuje nepřeberné množství příkazů z různých oblastí. Zde jsou poukázány jedny z nejběžnějších používající se při tvorbě odborných článků.

- `\usepackage{czech}` - přidává pomocné balíčky, jež obsahují pomocné funkce. Například přidání češtiny.
- `\author{John New}` - definuje jméno autora článku.
- `\title{Name article}` - definuje název článku.
- `\keywords{word1}` - definuje klíčová slova.

- `\maketitle` - příkaz určující vysázení.
- `\section{Kapitola}` - příkaz pro vytvoření nadpisu kapitoly.
- `\subsection{Podkapitola}` - příkaz pro vytvoření podnadpisu.
- `\tableofcontents` - příkaz automatické generování obrázku.
- `\cite` - příkaz pro přiřazení zdroje, odkud byly informace čerpány.
- `\includegraphics[scale=0.6]{pic/fig1.png}` - příkaz pro vložení obrázku se speciálním nastavením.
- `\begin{thebibliography}, \end{thebibliography}` definuje sekci, ve které se nachází reference.
- `\begin{\footnote{poznámka}` - příkaz pro vytvoření poznámky pod čarou.
- `\verb {zdrojovy kod}` příkaz pro výpis zdrojových kódů. Ignoruje tedy všechny speciální znaky a příkazy. A jsou vysázeny jako prostý text.

3.4.5 Tvorba vlastních příkazů

Při tvorbě složitějších projektů psaných v \LaTeX u, často dochází k situaci, kdy \LaTeX už nenabízí autorovi potřebný příkaz pro požadovanou speciální strukturalizaci. Řešením takového problému je definovat si vlastní příkazy.

U klasického použití příkazu se využívají 2 argumenty *Name* a jeho *definition*. Třetí argument *number* je volitelný a využívá se, pokud i tvořený příkaz má využívat argumenty (může nabývat hodnot 1-9) [2]. \LaTeX nedovolí přepsat příkaz, který již existuje, pro takové případy je potřeba použití příkazu `\renewcommand`, který používá stejnou syntaxi zápisu jako `\newcommand`.

```
\newcommand{name}[number]{definition}
```

Následující výpis ukazuje použití příkazu `\newcommand` s využitím i rozšířeného třetího argumentu v hranatých závorkách. V první části je vytvořený `\novyprikaz` s textem *novy prikaz* a definovaný vstupní argument, který bude vyznačen kurzívou.

```
\newcommand{\novyprikaz}[1]{ Novy prikaz \emph{#1}}
% v tele dokumentu
\novyprikaz{je na svete!}
```

Výsledný text bude poté vypadat: Novy prikaz *je na svete!*

4 Bash

Protože většina celého systému je vytvořena pomocí bash bude tato kapitola obsahovat popis vybraných bash příkazů a programových konstrukcí, které se v této práci vyskytují. Jedná se především o příkazy k práci s textovými soubory, tvoření podmínek a cyklů a podobně.

Bash je jeden z nejpoužívanějšího shellu v Linuxu. Shell je program, který v unixových operačních systémech vytváří základní textové rozhraní mezi uživatelem a operačním systémem. Jinými slovy interpretuje příkazovou řádku [6].

4.1 První skript

Celý skript je vlastně textový soubor, který obsahuje seznam příkazů. Tyto příkazy jsou poté analogicky prováděny. Je dobré vytvářenému souboru přidat příponu `.sh`. Není to však nutností, systém sám dokáže poté rozpoznat zda se jedná o skript.

Na první řádek skriptu se vždy napíše `#!/bin/bash`, jenž určuje jaký interpret se má spustit. Poté již, můžou následovat jednotlivé příkazy. Je dobré každý příkaz uvádět na samostatný řádek, pokud uživatel chce mít více příkazů na řádku, musí je oddělit středníkem[10]. Může nastat situace, že psaný příkaz je příliš dlouhý. Pro jeho rozdělení se používá znak `\`. Na spodním výpisu je ukázka jednoduchého skriptu. Tento skript se poté spouští v terminálu pomocí příkazu `.prvni_skript.sh` a vytiskne na obrazovku *Prvni skript*.

```
#navez soboru je prvni_skript.sh
#!/bin/bash
echo "Prvni skript"
```

4.1.1 Proměnné

V bashi stejně jako v jiných programovacích jazycích se používají proměnné, sloužící pro ukládání určitých hodnot a práci s nimi. Bash definuje proměnou pomocí dolaru `$`. Tedy krom prvního použití, kdy proměnné přiřazujeme nějakou hodnotu [10]. Ukázka skriptu s proměnnou je na výpisu č. ???. Proměnná A se uloží do proměnné B a ta se poté vypíše na obrazovku [10].

```
#!/bin/bash
A=1
B=$A
echo "Hodnota A promene je:" $B
```

4.1.2 Rozhodovací struktury

Aby mohlo být ovlivněno chování programu je potřeba definovat určité podmínky. Podmínky se v bashi definují pomocí příkazu `if`. Jak je vidět z výpisu č. ?? syntaxe příkazu je velmi podobná ostatním programovacím jazykům. Rozdíl tvoří uzavírací příkaz `fi`, který ukončuje podmínku. Většinou se s příkazem `if` píše na stejný řádek i `then` oddělený středníkem. Pokud by `then` bylo napsáno až na dalším řádku, středník by nebyl potřeba. Při složitějším větvení podmínky je možno použít zkrácenou verzi `else if`, a to `elif`. Nebo naopak `else` úplně vynechat [10].

```
if [ podminka ]; then
    prikaz
elif [ podminka2 ]; then
    prikaz
else
    prikaz
    prikaz
fi
```

Podobnou funkci jako příkaz `if` má příkaz `CASE`. Ten se většinou používá, pokud testovaná hodnota může vyhovovat více podmínkám. Důležité je za posledním příkazem každé sekce přidat dva středníky aby se rozlišilo, konec sekce od příkazu [10]. Na výpisu je ukázka `CASE` příkazu, který vyhledává hodnoty A nebo B. Pokud je hodnota jiná, je rozhodování ukončeno.

```
case $ hodnota in
    A) odpoved=$ hodnota;;

    * ) echo Zadalí jste spatnou hodnotu!!
        exit 1;;
    B) odpoved=$ hodnota;;
esac
```

4.1.3 Cykly

Pokud programátor potřebuje použít některé příkazy několikrát za sebou, může je umístit do cyklu. Existují 3 druhy cyklů. Je to `UNTIL`, který říká, dokud není něco hotové, dělej. Zatím co `WHILE` určuje podmínku, jestli to platí, udělej to. Třetím příkazem je příkaz `for`. U kterého definujeme přesný počet cyklů, jenž se má vykonat. Níže jsou uvedeny syntaxe všech tří zmíněných cyklů.

```
while/until [ podmínka ]  
do  
    příkazy  
done
```

```
-----  
for promenna in seznam  
do  
    příkazy  
done
```

4.2 Vybrané bash příkazy

V této kapitole jsou vybrané příkazy pro práci se soubory, které se poté vyskytují ve vytvořeném systému pro automatizaci tvorby vydání časopisu.

4.2.1 SED

Jedná se o neinteraktivní textový editor, který čte řádky ze standardního vstupu a provádí s nimi zadané příkazy, výsledek pak zobrazuje na standardním výstupu. Sed podporuje i regulární výrazy. Umožňuje například změnit slova v textu. Na ukázce pomocí se změni slovní spojení zelený dům za modrý dům [9].

```
echo "zeleny dum" | sed 's/zeleny/modry/'
```

4.2.2 HEAD

Slouží primárně k zobrazení prvních několika řádků souboru. Následující příkaz vypíše prvních 5 řádků souboru [8].

```
head -5 soubor.txt
```

4.2.3 TAIL

Tail je obdobný příkazu head, ale na rozdíl od něj funguje opačně, tedy zobrazuje posledních několik řádků v souboru. Příkaz vypíše posledních 10 řádků souboru [8].

```
tail -10 soubor.txt
```

4.2.4 CUT

Je program sloužící k vytažení určitých částí z textu [7]. Mezi nejdůležitější přepínače patří `-d` (oddělovací znak) a `-f` (určení polí, která se mají vypisovat). Na výpisu může pozorovat příkaz, který vypíše 6 sloupec ze souboru ve formátu CSV oddělovaný čárkami.

```
cut -d ',' -f 6 soubor.csv
```

4.2.5 TR

Čte data ze standardního vstupu, transformuje je dle zadání a posílá na standardní výstup. Díky přeměrování vstupů a výstupů umožní snadno zpracovávat soubory [9]. Například pomocí `tr` lze zaměnit v souboru všechny „e“ za „o“.

```
cat /etc/soubor.txt | tr "e" "o"
```

4.2.6 SORT

Sort je program, jehož účelem je seřadit řádky daných souborů. Na příkladu příkaz vybere prvních 5 řádků souboru.txt a setřídí je [7].

```
head -n 5 soubor.txt | sort
```

4.2.7 AWK

Awk je také často používaný program pro práci s textem. Příkaz je ovlivněn sérií pravidel, která mohou obsahovat vzor, akci nebo obojí. Akce je uzavřena do `{ }`. Když je ve vstupu nalezen vzor, vykoná se příslušná akce. Příklad zobrazí všechny řádky obsahující slovo ahoj [9].

```
awk '/ahoj/ {print}' /home/dopis.txt
```

4.2.8 CAT

Příkaz `cat` slouží ke spojení nebo vypsání obsahu souborů (s libovolným obsahem) [8]. Spojení několika souborů (v daném pořadí) do jednoho se může využít následující příkaz.

```
cat soubor.txt soubor2.txt soubor3.txt > velky_soubor.txt
```

5 Tralics

Tralics je nástroj, který byl původně navržen pro projekt RAWEB⁶. Tento skript původně napsaný v Perlu se vyvinul do podoby L^AT_EX-to-XML převaděče. Poté byl přepsán do jazyka C++ a přejmenován na Tralics. Aktuální verze se používá jak pro PDF tak HTML verze.

Hlavním úkolem tohoto programu je číst a analyzovat vstupní dokument, řídit (zjednodušit, zkompletovat, setřídít) a vytvořit nový dokument. Tralics je multiplatformní, kromě zdrojových textů jsou k dispozici i předpřipravené zkompilované binární verze tohoto nástroje pro operační systém GNU/Linux, Apple Mac OS X a Microsoft Windows [15].

5.1 Struktura L^AT_EXového dokumentu

Tralics předpokládá, že dokument, který má být přeložen, splňuje některé L^AT_EXové normy. Dokument obsahuje příkaz `\documentclass`, dále následující řádky, které obsahují `\usepackage` příkazy, následující prostředí dokumentu. Na výpisu je jednoduchý příklad L^AT_EXového souboru a pod ním obsah výsledného XML souboru.

```
\documentclass{article}
\begin{document}
    Dokument je na svete!
\end{document}
```

```
-----
<?xml version='1.0' encoding='iso-8859-1'?>
<!DOCTYPE std SYSTEM 'classes.dtd'>
<!-- Translated from latex by tralics 2.13.0,
      date: 2014/02/22-->
<std>
    <p> Dokument je na svete!</p>
</std>
```

Jak je z předchozího výpisu patrné, XML bylo vytvořeno. Obsahuje hlavičku udávající verzi XML, znakové kódování (tato vlastnost se dá pomocí přepínačů nastavit viz kapitola 5.3. Dále je uvedený soubor DTD. Podle kterého je ovlivněna vizuální podoba výsledného XML souboru. Následuje sekce s verzí tralicsu a datem vytvoření dokumentu. Poté následuje samotné tělo XML souboru obsahující jednotlivé elementy.

⁶RAWEB je experimentální aplikace, jenž slouží pro prohlížení v výzkumných reportech z týmu NNDRIA

Ovšem při složitějších \TeX ových souborech, by struktura výsledného programu nebyla optimální, proto se při překladu přidávají speciální konfigurační soubory, které upraví výstup podle toho, jak si ho nadefinujeme.[13]

5.2 Konfigurační soubory

Konfigurační soubory jsou používány k ovlivnění výstupu Tralicsu. Základní konfigurační soubor nebo specializované soubory s příponou TCF se používají pro úpravu některých prvků, popřípadě DTD⁷ v XML. Další soubory s příponami Ult, PLT a CLT, lze použít namísto balíčků nebo třídních souborů. Ult je zkratka pro uživatele, ostatní písmena jsou pro \LaTeX a Tralics.[11]

- Příkaz `tralics-noconfig`, nenačte žádný konfigurační soubor.
- Příkaz `tralics-ConfigFile = foo`, pak Tralics vypíše že zkusí konfigurační soubor z uživatelských specifikace, a pokusí se ho použít.
- Při použití `ConfigFile = foo.tcf` Tralics vypíše to co bylo uvedeno v předchozím bodě a bude hledat soubor v adresáři "confdir".

5.2.1 Ukázka dokumentu s konfiguračním souborem

Mějme konfigurační soubor *example.tfc* jenž má danou strukturu.

```
DocType = hello world.dtd
BeginAlias
    std report book article minimal
End
BeginCommands
    \def\world{world}
End
```

Příkaz `docType` mám určuje, jakým způsobem budou výsledné data v XML souboru prezentovaná. Tyto instrukce jsou uvedeny v dokumentu s příponou *dtd*. Tento konfigurační soubor spustíme společně se \LaTeX ovým dokumentem, který má strukturu uvedenou ve výpisu.

```
\documentclass{article}
\def\hello{\uppercase{N}ovy }
\begin{document}
    \hello{} dokument s pomoci konfiguracniho souboru
\end{document}
```

⁷DTD , česky Definice typu dokumentu) je jazyk pro popis struktury XML

Spuštěním příkazu :

```
tralics soubor.tex - configfile = example.tcf
```

Dostaneme takto strukturované XML:

```
<?xml version='1.0' encoding='iso-8859-1'?>
<!DOCTYPE std SYSTEM 'classes.dtd'>
<!-- Translated from latex by tralics 2.13.0, date: 2014/02/22-->
<std>
    <p> Novy dokument s pomoci konfiguracniho souboru</p>
</std>
```

5.3 Přepínače pro Tralics

Aby se povedlo převést L^AT_EXový soubor do XML, je potřeba nějakým způsobem tento proces spustit. V linuxovém prostředí se spouští pomocí příkazu `tralics` v terminálovém okně. K tomuto se dá připojit velké množství nastavení. Některé z nich zde budou uvedeny. [12]

- `confdir=prefix` určuje místo, kde se nachází konfigurační soubor
- `config = FILE` použití souboru `FILE` namísto výchozího konfiguračního souboru
- `help` vypíše nápovědu
- `input_dir = DIR` specifikuje, kde se nachází zdrojový soubor. Může obsahovat několik míst oddělených čárkou
- `input_file = FILE` určuje, který `FILE` se má přeložit do XML (je možnost přidat příponu `.tex`)
- `latin1` říká, že vstupní soubory jsou standardně zakódovány v `iso-8859-1`
- `noconfig` nebude použita žádná konfigurace
- `output_dir = DIR` určuje místo, kde se uloží vytvořený XML soubor
- `silent` příkaz pro zmenšení výpisu na obrazovku při převodu
- `verbose, -v` příkaz pro podrobnější výpis při převodu
- `oe8, -oe1, -oe8a, -oe1a` určuje kódování výsledného XML. Může být buď UTF8 nebo `latin1`

5.3.1 Tralics a matematické výrazy

V některých případech potřebujeme z \LaTeX ových souborů dostat matematické výrazy. Tralics automaticky převádí tyto výrazy do jazyka MathML⁸ [14]. Pro ukázkou byla zvolena Pythagorova věta definována souborem:

```
\documentclass{article}
\begin{document}
  Pythagorovu vetu vyjadruje tato rovnice:
  \[a^2 + b^2 = c^2\]
\end{document}
```

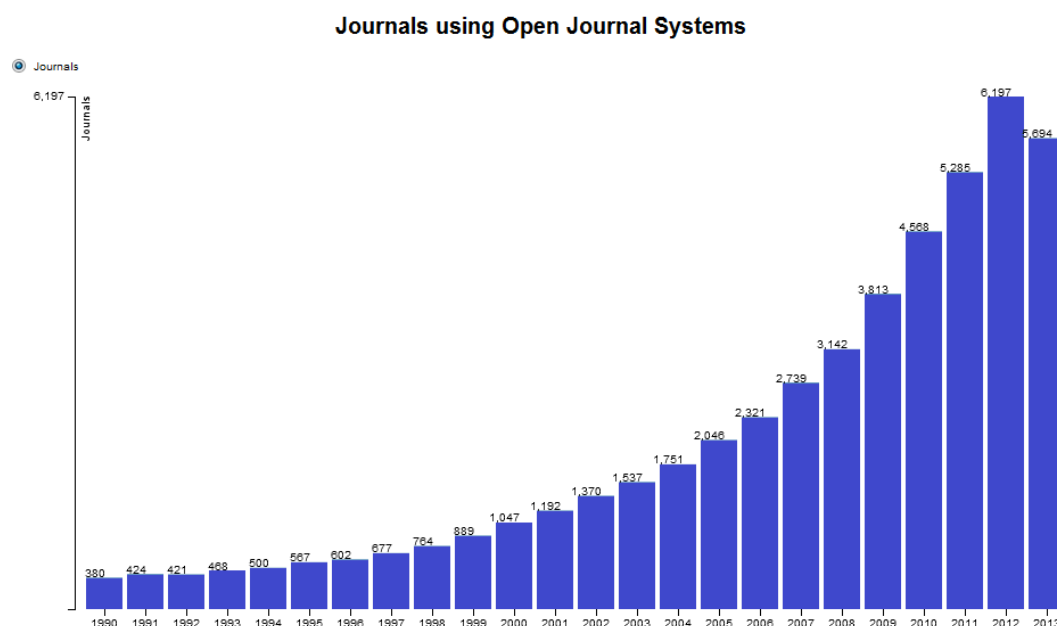
Po zpracování výše uvedeného souboru pomocí tralicsu je výsledné XML vygenerováno do dané podoby:

```
<?xml version='1.0' encoding='iso-8859-1'?>
<!DOCTYPE std SYSTEM 'classes.dtd'>
<!-- Translated from latex by tralics 2.13.0, date: 2014/02/22-->
<std>
<p>Pythagorova veta:</p>
<formula type='display'>
<math mode='display' xmlns='http://www.w3.org/1998/Math/MathML'>
  <mrow><msup><mi>a</mi> <mn>2</mn> </msup>
    <mo>+</mo>
    <msup><mi>b</mi> <mn>2</mn> </msup>
    <mo>=</mo>
    <msup><mi>c</mi> <mn>2</mn> </msup>
  </mrow>
</math>
</formula>
<p noindent='true' />
</std>
```

⁸MathML je nízkoúrovňová specifikace pro popis matematiky jako základu, který je potřebný pro zařazení matematických výrazů na webové stránky

6 Open Jurnal System

OJS je systém pro správu a publikování časopisů, který pomáhá při všech fázích procesu publikování. Od podání až po on-line publikaci a indexaci. Podporuje funkce jako vyhledávání, editaci a prohlížení článků. Open Jurnal System prošel několika fázemi vývoje. Poslední vydaná verze je 2.3.1. Ale její jádro je založeno na stabilnější verzi 2.2.4. Tento software je open source a je volně šiřitelný pod GNU General Public License. Tento systém patří k velice oblíbeným po celém světě [16]. Jak je možné vidět na obrázku č. 3. Graf ukazuje vzrůstající počet webových časopisů používající tento redakční systém.



Obrázek 3: Graf používání OJS ve světě [17]

6.1 Funkce OJS

Aby bylo možno podrobně pochopit fungování tohoto systému, je potřeba poznat jednotlivé procesy při tvorbě časopisu. Přesnější popis se nachází na obrázku č 4. Celý proces se skládá z 5 hlavních kroků. První je podání článku, druhý je přezkoumání podání, úprava podání správa čísla a samotná publikace. Uvedené procesy probíhají i při tvorbě časopisu AEEE. Protože OJS podporuje více rolí jako například manažer časopisu, autora, editora. Registrovaní uživatelé mohou zastávat, některé z těchto rolí a plnit jejich speciální úkoly. OJS souvisí především s třetím krokem procesu při tvorbě časopisu. V tomto kroku musí layout editor sestavit kolekci článků v daných formátech jako HTML, PDF PS a jiné. Aby poté redaktori mohli vydat finální elektronickou publikaci.

6.2 Instalace OJS

Podmínkou pro správné fungování je mít na webovém serveru jsou nainstalovány tyto programy [18]:

- PHP (4.2.x nebo vyšší verze)
- MySQL (3.23 vyšší verze) případně PostgreSQL (7.1 nebo vyšší verze)
- Apache (1.3.2x nebo vyšší verze) případně Apache 2 (2.0.4x nebo vyšší verze) nebo Microsoft IIS 6 (s PHP 5.x)
- Operační systém Linux, BSD, Solaris, Mac OS X, Windows)

Samotná instalace OJS je poměrně snadná a rychlá, stačí z webových stránek⁹ stáhnout komprimovaný archív a rozbalit jej do složky webového serveru na Linuxu. To ve většině případů bývá umístění */var/www/*.

Po nakopírování, je nutné si upravit v Linuxu práva pro některé soubory a složky. Za pomoci terminálu použijeme následující příkazy uvedené ve výpisu. Čísla označují druh práv, které jsou použity na jednotlivé složky či soubory. Čísli 777 nastavuje, že adresář je volně přístupný. Podobné pravidlo určuje i 666 pro soubory.

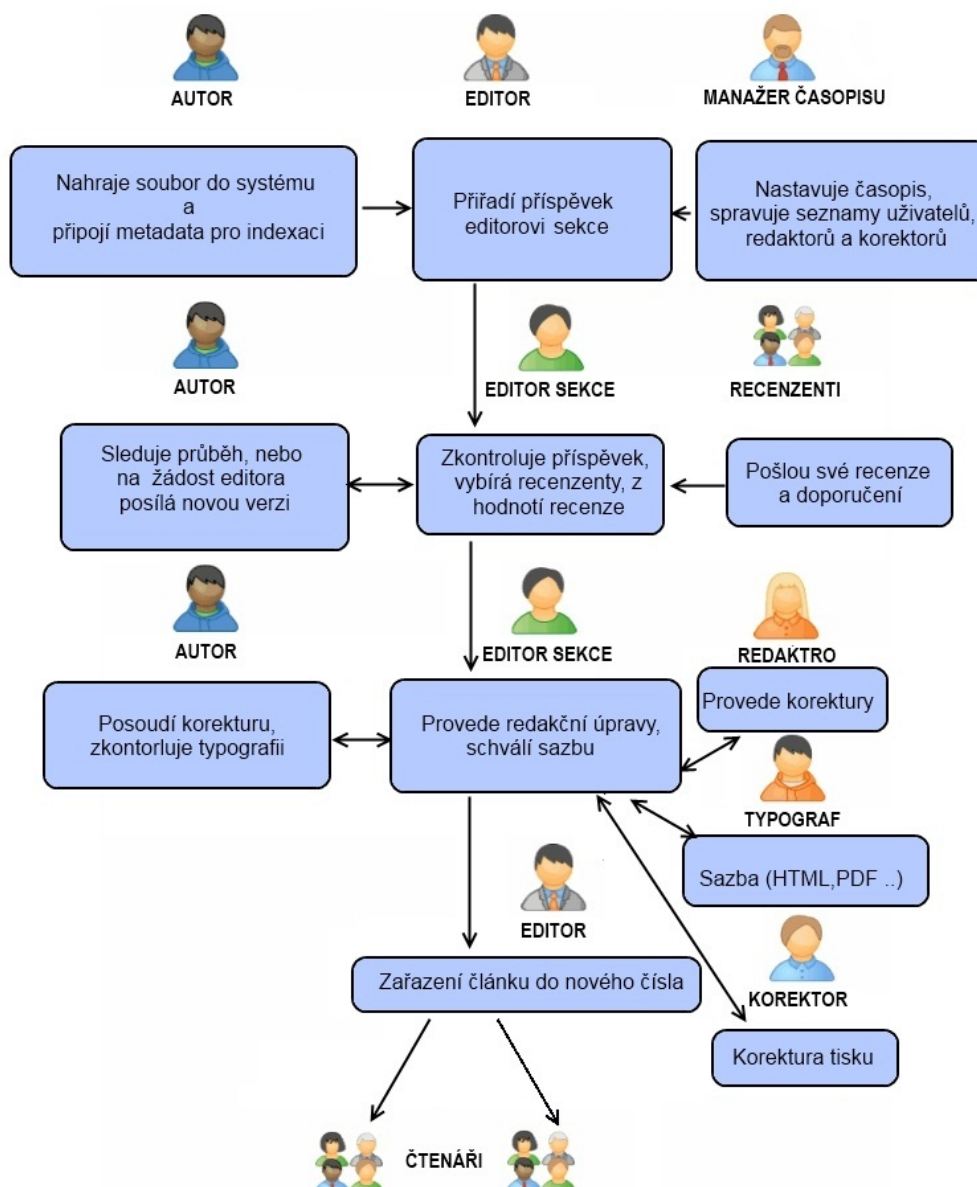
```
chmod 666 config.inc.php
chmod 777 public/ -R
chmod 777 cache/ -R
chmod 777 CS
```

Poté stačí, otevřít webový prohlížeč a napsat do adresy *localhost/ojs*. Stránka se přesměruje na stránku, kde proběhne dokončení instalace systému. Zde se nacházejí nastavení jazyků, možnosti šifrování, nastavení přístupu do databází a mimo jiné také se zde nastavuje adresář pro soubory, které budou nahrávány na server uživateli. Pro tuto složku je současně nutno upravit práva. Viz poslední příkaz z výpisu č.???. Nesmí se zapomenout také na nastavení administrátorského účtu.

6.3 Role OJS

OJS má velkou škálu rolí. Tím umožňuje rozdělit práci mezi uživatele a rozdělit jim pracovní postupy, popřípadě omezit přístup do různých částí systému. Protože jedna instalace OJS může obsahovat několik časopisů, uživatelé mohou být zapsáni v různých rolích pro více časopisů najednou [19]. V následujících podkapitolách mé práce budou jednotlivé role vysvětleny.

⁹ <http://pkp.sfu.ca/ojs/ojs.download/>



Obrázek 4: Popis procesů probíhajících při tvorbě časopisu

6.3.1 Site Administrator

Administrátor je zodpovědný za celkovou funkčnost OJS. Zajišťuje nastavení serveru, umožňuje vytvářet nové časopisy, přidávat jazykové soubory, spravovat jednotlivé uživatelské účty, exportovat nebo importovat data či zapínat nejrůznější plugíny. Účet je automaticky vytvořen při instalaci. Na rozdíl od všech ostatních OJS rolí, může existovat pouze jeden správce webu [20].

6.3.2 Journal Manager

Manager Journal je odpovědný za vytvoření webové stránky časopisu, Má právo k nastavení systému a řízení uživatelských účtů. Dále umožňuje vyplňování webových formulářů a nahrávání souborů. Journal Manager také určuje uživatele pro role Section Editors, Copyeditors, Layout Editors, Proofreaders, Authors, and Reviewers. Případně je dokáže importovat do systému. Uživatelé z jiných databází. Tento uživatel má také přístup k dalším funkcím pro správu časopisu, a může vytvořit nové oddíly pro časopis, nastavit Review Formuláře, upravit výchozí e-maily, spravovat nástroje pro čtení, prohlížení statistik a reportů[21].

6.3.3 Reader

Čtenář je nejjednodušší role v celém systému. Má nejmenší počet možností. Jeho hlavní funkcí je aby uživatel pod touto rolí dostával zprávy o nově vydaných číslech časopisu. [29].

6.3.4 Author

Autoři nahrávají své články do časopisu přímo na internetových stránkách. Autor je vyzván, aby při nahrávání souboru článku poskytl metadata nebo informace pro indexování. Tento uživatel může také nahrát doplňkového soubory, ve formě datových souborů, výzkumných nástrojů nebo zdrojových textů. Takto registrovaný uživatel je schopen pozorovat jakými procesy jeho dílo v redakčním systému právě prochází. A souběžně je vyzván k redigování a korekci napsaného díla [22].

6.3.5 Editor

Editor dohlíží na kompletní proces publikování. Jinými slovy lze editora nazvat šéfredaktorem. Editor společně s Journal Managerem stanovují pravidla a funkce v časopisu. Editor se stará o hladký průběh mezi jednotlivými procesy publikování a pomáhá řešit problémy. A vytváří nové vydání časopisu a určuje, které články v něm budou publikovány[23].

6.3.6 Section Editor

Editor sekcí řídí přezkoumání a úpravy vložených článků od autorů, které jim byly přiděleny. V některých případech, tato role dohlíží pouze na procesy kontroly článků nebo je rovněž odpovědný za dohled na podání, které jsou přijaty v rámci procesu úpravy (to znamená, redigování, úpravy rozložení a korektury). Nicméně Editor sekcí často pracuje pouze v dohlížecím módu. A editor působící v roli editora sekce vidí podání během jednotlivých procesů úprav v časopisu [24].

6.3.7 Reviewer

Recenzent je vybrán Section Editorem k přezkoumání článku vloženého autorem. Recenzenti by měli poskytnout hodnocení budoucích publikovaných děl. Tyto informace by poté měli pomoci Editorovi a Autorovi k úpravě daného článku [25].

6.3.8 Subscription Manager

Role starající se o předplatné časopisu a platební politiku. Zařizuje všechny potřebné funkce ohledně předplatného [18].

6.3.9 Copyeditor

Copyeditor v systému zastává pozici, jenž má na starosti zlepšení gramatiky a srozumitelnosti publikovaných děl. Dává pozor na dodržování bibliografické a textové styly. Přípravuje dokumenty pro Layout Editory. U některých časopisů tuto funkci zastávají Editoři nebo Editoři sekcí [26].

6.3.10 Layout Editor

Editor rozvržení pracuje s verzí článku přezvanou od copyedited. Tedy soubory, které se časopis rozhodl použít pro elektronické publikace. Systém OJS neposkytuje software pro konverzi textových dokumentů do jiných formátů, takže Layout Editoři by měli mít přístup používání software třetích stran. Program by měl vytvořit čitelné, přehledné a dobře strukturované dokumenty vhodné pro publikaci vědeckých časopisů. V některých případech, editor nebo editor sekce mohou zastávat funkce jako Layout Editor [27].

6.3.11 Proofreader

Korektor pečlivě přezkoumává články, které upravil Layout editor. Snaží se hledat a zaznamenávat veškeré typografické chyby a formátování. Které poté nechá opravit. U některých časopisů, editor nebo editor sekce mají funkce jako korektor [28].

6.4 Přídavné moduly

Přídavné moduly umožňují OJS rozšířit jeho funkčnost. Mezi nejčastěji používané pluginy, které obsahuje základní instalace OJS ve výchozím nastavení. Journal Manager se můžete rozhodnout, které pluginy použije ve svém časopisu, a které vynechá. V současné verzi jsou zahrnuty pluginy jako ověřování uživatelů pomocí LDAP¹⁰, import a export v XML, nástroj pro indexování na Google Scholar, vzhledové motivy. A pro tuto práci využívá nejdůležitější modul reportování do formátu cvs. Kvůli snadnější tvorbě řídicího soboru viz. kapitola 7.5.4

¹⁰Protokol pro přístup k datům z adresářového serveru.

7 Systém pro automatickou tvorbu časopisu

V této kapitole a jejích podkapitolách uvádím podrobný popis vytvářeného systému. Popisuji zde způsob vytváření a fungování systému, jeho adresářovou strukturu i jednotlivé pomocné soubory, jenž vstupují do procesu tvorby celého čísla. Některé podkapitoly se podrobně věnují jednotlivým pomocným souborům i samotným skriptům.

7.1 Ruční tvorba časopisu

Vkládaný článek musí projít velkým množstvím procesů, než se dostane do finální podoby. Tento vytvořený systém by měl ulehčit práci editorům časopisu a minimalizovat nevýhody vyskytující se při ručním zpracování. Při ručním sestavování je potřeba každé stránce článku zvlášť upravit záhlaví a zápatí. Nastavit číslování stránek. Záhlaví a zápatí se musí měnit také u zbylých stránek časopisu. Tyto listy obsahují doplňující informace k danému vydání. Jako je obsah, který se musel ručně vytvářet, nebo úvodní slovo autora a mnoho dalších. Tím, že se do těchto částí musí nesčetněkrát zasahovat, je tento postup velice zdoluhavý, a je lehké zde kdekoliv udělat chybu, což prodlužuje tvorbu celého čísla.

7.2 Možnosti řešení

Vymyšlení celého postupu sestavování časopisu patřilo k nejtěžším částem mé diplomové práce. Bylo důležité správně zvolit použité technologie. Při prvním pohledu na tuto problematiku se naskytly dvě základní cesty.

Jednou z cest bylo využití samotné technologie \LaTeX . A pomocí ní vytvořit speciální třídy a supertřídy, které by celý proces sestavování řídily. Druhá cesta se skládá ze sestavování časopisu pomocí bash skriptů, pomocí kterých by se spouštěly jednotlivé funkce.

Pro tuto práci jsem zvolil cestu pomocí bash skriptů. Bylo to především z důvodů, že implementace pomocí samotného \LaTeX u by byla náročnější pro méně zkušené uživatele, a složitější konstrukce by nemusely být pro ně dostatečně čitelné. Také pokud bychom museli pracovat s nějakými přídatnými balíčky, které neobsahují všechny verze \LaTeX u, mohly by zde vzniknout problémy. Bash naopak je více využívaný pro podobné implementace než \LaTeX .

7.3 Schéma systému

Celý proces tvorby časopisu je znázorněn na obrázku č. 5, kde vidíme, jak je daný systém strukturovaný. Celý program se spouští skriptem `start.sh`, ten poté volá další skripty, které postupně z jednotlivých částí vytvoří jeden dokument. Celý proces se dá rozdělit do několika fází. V první fázi se vybere, které číslo časopisu se bude skládat. Určuje se podle roku a čísla vydání, jenž definuje uživatel. Poté proběhne kontrola, zda ve složce s vydáním jsou obsaženy všechny potřebné soubory a to: `source.csv`, `about.txt`,

`foto.jpg` a `predni_strana.pdf`. Pokud některý z daných dokumentů není obsažen v adresáři vydání, je program ukončen s hláškou o nenalezení některého ze souborů.

V další fázi se vytvářejí úvodní listy časopisu. Jde o listy, které mívají ve všech číslech stejný obsah, pouze se zde mění záhlaví a zápatí stran. Výjimku tvoří stránky s obsahem a s úvodním slovem. Tyto stránky jsou také tvořeny dynamicky za pomoci vstupních souborů `about.txt` a `foto.jpg` se vytvoří úvodní slovo a z informací ze souboru `source.csv` se vytvoří obsah. Takto přeložené stránky již do formátu PDF jsou spojeny do dvou souborů připravených na další zpracování.

Třetí částí celého procesu je vytvoření samotných článků, které jsou již po korektuře a úpravách, připraveny pro začlenění do časopisu. Těmto listům se přidává záhlaví a zápatí, plus z každého článku je vytvořený pomocný soubor `pomoc.tex` sloužící pro program Tralics, který z něj vytvoří XML dokument obsahující metadata publikovaných článků sloužící pro indexaci do digitálních knihoven. Takto vytvořený soubor je posléze přiložen k vytvořenému článku. Následně jsou všechny články seskupeny do jednoho souboru.

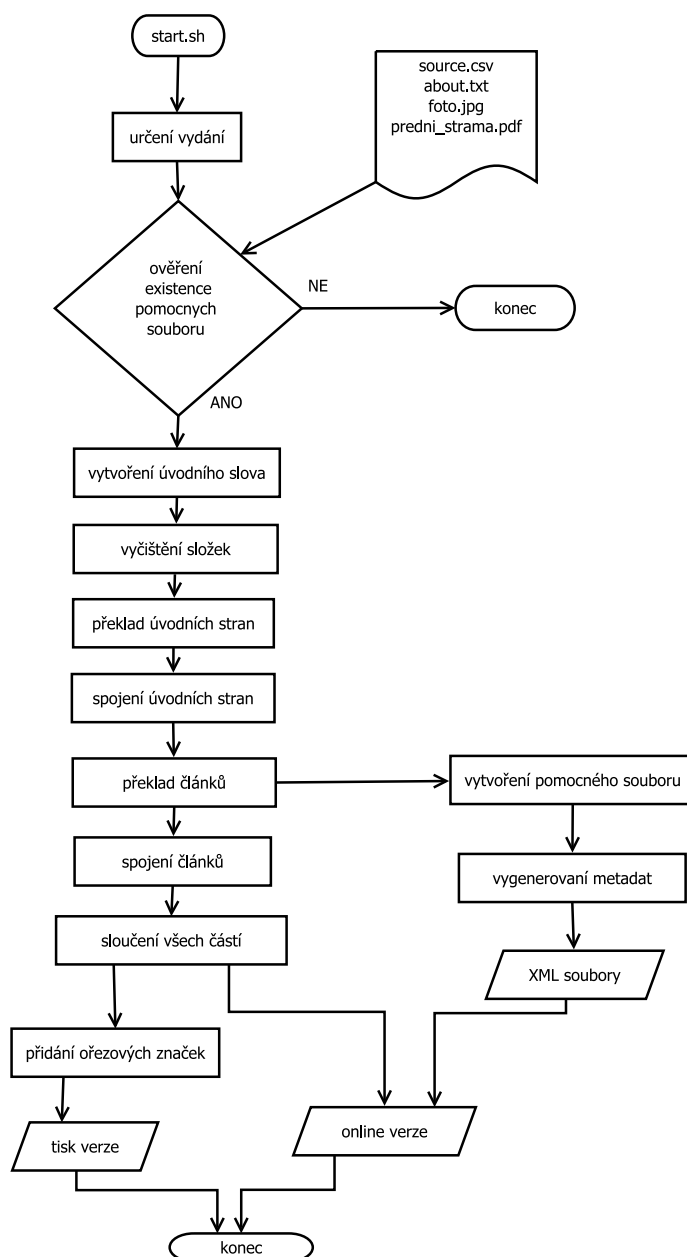
V poslední fázi proběhne spojení dokumentů vytvořených v předchozích fázích a rozdělení vytvářeného časopisu nad dvě verze. Verzi pro tisk a verzi pro online publikaci. Verze pro tisk projde poté posledním procesem celého systému, kde jsou ji přidány ořezové značky.

Po dokončení celého skriptu jsou tedy připraveny dvě různé verze. Je to verze pro tisk umístěná ve složce `tisk`, a verze pro online publikování, která společně se samotnými články a XML soubory je umístěna ve složce `online`.

7.4 Adresářová struktura systému

Celý systém, má pevně stanovenou souborovou strukturu, která se musí pro správnou funkčnost zachovávat. Při obměně starších souborů za aktuální je důležité dávat si pozor na velké a malé písmena. Jejich záměna může způsobit opět špatnou funkčnost. Nicméně celá struktura je nadefinována přehledně, aby nemělo by zde docházet ke zbytečným komplikacím.

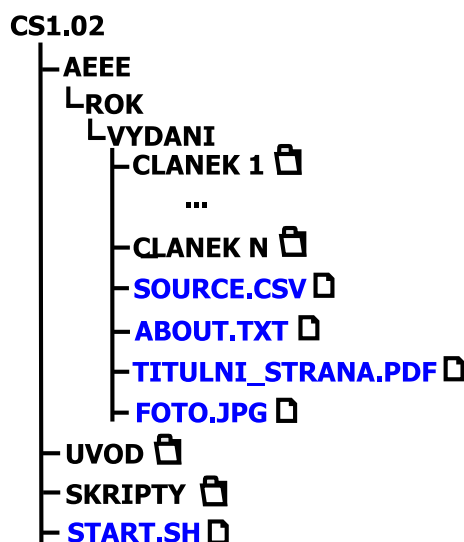
Výše zmíněná struktura je znázorněna na obrázku č. 6. Černě jsou zobrazeny složky a modře jednotlivé soubory. Celkově se v kořenovém adresáři CS1.02 nacházejí 3 složky (`AEEEE`, `SKRIPTY`, `UVOD`) a hlavní spouštěcí skript `start.sh`, kterým se celý systém tvorby spouští. Složka `UVOD` obsahuje úvodní listy časopisu, Úvodní listy se nachází u každého vydání a potřebují minimální úpravu. Dalším adresářem je složka `SKRIPTY`, která obsahuje ostatní bash skripty pro vytvoření časopisu a další pomocné soubory, jenž jsou důležité pro jeho tvorbu. Třetí složka obsahuje samotný časopis. Jejimi podsložkami jsou jednotlivé roky a ty obsahují čtvrtletní vydání časopisu. V každém vydání se nacházejí jednotlivé články, které dané vydání bude obsahovat. Mimo články obsahuje složka vydání soubory `about.txt` obsahující úvodní slovo. `Foto.jpg`, `titulni_strana-pdf` a soubor ve formátu CSV, co obsahuje veškeré informace o daném vydání. Jednotlivé složky a soubory budou podrobněji popsány v následujících kapitolách.



Obrázek 5: Schéma systému

7.5 Popis jednotlivých částí systému

V následujících podkapitolách bude uveden, podrobnější popis jednotlivých složek a souborů, jenž vstupují do procesu tvorby časopisu. Nejprve budou popsány jednotlivé složky a poté jejich dokumenty, které obsahují. Výjimku tvoří zdrojový soubor `source.csv`, ježmuž bude věnovaná samostatná kapitola, jelikož se jedná o jeden z nejdůležitějších prvků.



Obrázek 6: Souborová struktura systému

7.5.1 Složka DESKY

Tento adresář se nachází v adresáři UVOD a obsahuje 8 podsložek. Každá tato podsložka reprezentuje jeden list úvodních stran časopisu. U listů, kde neprobíhají žádné úpravy je obsah složky následující:

- DESKA_xx.tex \LaTeX ový soubor s danou stránkou
 - Headfoot.tex \LaTeX ový soubor se záhlavím a zápatím
 - AEEE.cls třída definující vzhled stránky
 - Logo.jpg obrázek vkládající se do záhlaví stránky
 - Pagex.pdf Vzorová strana vkládající se pro překlad do DESKA_xx.tex
- x zastupuje číselné hodnoty.

Některé složky neobsahují všechny tyto soubory. Je to dáno tím, že obalové listy časopisu nepotřebují vkládat záhlaví a zápatí. Zde je důležité, aby název \LaTeX ového souboru se shodoval s názvem dané složky. Je to proto, že při překladu se skript orientuje právě podle tohoto názvu. V každém z těchto souboru se importuje PDF s již vytvořenou stránkou, u které se pouze ořežou staré záhlaví a zápatí a doplní se o nové. K tomuto se

používá sada příkazů, které ve svém nastavení z každé strany ořezou stránku o několik centimetrů. Ořez probíhá v pořadí: zleva, z vrchu, zprava a zespodu. Velikost ořezání u jednotlivých stánek úvodních listů se může lišit. Je to z důvodu, aby se stránka co nejvíc podobala originálu.

```
\begin{figure}[!htbp]
\centering
\includegraphics[trim=23mm 2.8cm 10mm 2.8cm,clip]{page3.pdf}
\end{figure}
```

Speciální funkci mají složky DESKA_V a DESKA_VI.

7.5.1.1 DESKA_V obsahuje úvodní slovo. Text je na tuto stránku vkládán pomocí speciálního skriptu (kapitola č. 7.6.2) ze souboru `about.txt`. Kromě samotného textu je zde přidávána fotka autora, jenž psal úvodní slovo. Na obrázku č. 7) můžeme vidět výřez z PDF souboru vytvořeného pomocí DESKA_V.



VOLUME: 11 | NUMBER: 4 | 2014 | FEBRUARY

Foreword about Dr. Anirban Dhar, Scientist of Fiber Optics & Photonics Division, CSIR-Central Glass & Ceramic Research Institute, Kolkata, India:

Anirban Dhar was born in Kolkata, India in 1977. He graduated in Chemistry from Calcutta University in 2001. Anirban started his research work on fabrication of rare-earth doped specialty optical fiber and received his Ph.D degree in Chemistry from Jadavpur University, Kolkata, India in 2008. He then joined in Optical Fiber Technology Division, Institute of Photonics & Electronics, ASCR v.v.i, Prague, Czech Republic as postdoctoral fellow in 2009 and was involved in many important projects in different capacity as member or as principal investigator. In August 2010, he moved to Optoelectronics Research Centre, University of Southampton, UK and was involved in many important EPSRC and industry funded projects. Anirban joined as Scientist in Fiber Optics & Photonics Division of CSIR-CGCRI in May, 2012 and presently responsible for development of new varieties of specialty optical preform/fiber. Anirban is a member of OSA and acting as



John Clever

Obrázek 7: Ukázka úvodního slova daného vydání

7.5.1.2 DESKA_VI vytváří automaticky obsah časopisu, který obsahuje název článku, stránku a autory. Obsah je generován v samotném L^AT_EXovém souboru, kde za pomocí balíčku `datatool` je načtený csv soubor `zdroj.csv`. Z něj jsou pak načítány sloupce autoři, `start_page` a název článku. Ukázku generovaného obsahu lze vidět na obrázku č.8.

Kromě těchto složek se zde nachází také pomocný adresář CELEK, do které se poté shromažďují jednotlivé články převedené do PDF.



Contents

<i>Fault Indicators of Partial Discharges in Medium-Voltage Systems</i> Jan Moravec, Petr Novák	56
<i>Statistical Analysis of Compression Methods for Storing Binary Image for Low-Memory Systems</i> Jakub Fidler, Jiří Mops, Andreas del Potro, Petr Novák	61

Obrázek 8: Ukázka obsahu daného vydání

7.5.2 Složka skripty

Adresář se nachází v kořenové složce a obsahuje dodatečné skripty pro tvorbu časopisu a velké množství pomocných souborů. Tyto soubory jednotlivě vstupují do procesu tvorby. Jednotlivé skripty budou samostatně popsány v kapitole 7.6, aby lépe vynikly souvislosti mezi jednotlivými procesy. Každý pomocný soubor, je vždy nakopírován do určitého adresáře, kde je zpracován pomocí bash skriptu a po použití opět smazán. A zde jsou popsány jednotlivé pomocné soubory.

7.5.2.1 Aeee.tcf Je konfigurační soubor pro program Tralics převádějící L^AT_EXové dokumenty na XML soubor. V `Aeee.tcf` se definují jednotlivé elementy, které má výsledné XML obsahovat. Zde lze upravovat i konečný vzhled. Konfigurační soubor je používán při vytváření metadat z jednotlivých článků. Výsledné XML poté slouží pro možnost importování metadat do digitálních knihoven. Na ukázce je část konfiguračního souboru, kterým se definují jednotlivé elementy. Pokud tyto elementy nejsou vyplněny, jsou nahrazeny výchozím textem, uvedeném v uvozovkách.

```
BeginTitlePage
  \Author <Author> "No author given"
  \Title <Title> "No Title given"
  \Abstract <Abstract> "No Abstract given"
  \KeyWords <KeyWords> "No KeyWords given"
End
```

7.5.2.2 Blank.pdf Soubor s prázdnou stránkou ve formátu PDF. Slouží při vytvoření tiskové verze. Pokud skript zjistí, že celý časopis má lichý počet stran, přidá na předposlední místo prázdnou stránku.

7.5.2.3 Casopis.tisk.tex L^AT_EXový dokument, který je využit pro přidání ořezových značek. Při vytváření verze pro tisk. Jsou zde použity balíčky `Geomety` a `Crop`. Ve vlastnostech těchto balíčků jsou poté nastaveny parametry jako šířka a výška papíru a velikost

stránky. Zde byla zvolena velikost A3, protože samotný časopis je v přesné velikosti A4, takže by se ořezové značky neměly kde zobrazit. Pokud by tento styl ořezových značek nebyl dostačující je možno použít speciální balíček `zwpagelayout`¹¹, jenž byl přímo navržen pro tyto účely.

```
\usepackage[paperwidth=210mm,paperheight=297mm]{geometry}
\usepackage[cam,a3,center,pdftex]{crop}
```

7.5.2.4 Deska_V_vzor.tex Dokument, který slouží jako šablona při vytváření úvodního slova. Šablona obsahuje kostru celé stránky a v ní na určitých místech definovaná klíčová slova, která jsou poté za pomoci skriptu (kapitola č. 7.6.2 nahrazována textovými řetězci. Jedná se o klíčová slova `AUTOR`, `ZPRAVA`, `NADPIS`. Na ukázce můžeme vidět část kódu s obsahujícím klíčovým slovem `AUTOR`.

```
\begin{center}
\fcolorbox{seda}{seda}{\includegraphics[width=44mm]{foto.jpg}}
\captionof*{figure}{\sffamily\AUTOR}
\label{obr:zleva}
\end{center}
```

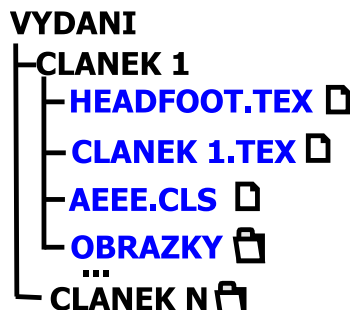
7.5.3 Složka AEEE

Adresář AEEE je, složku obsahující všechny edice časopisu. Jsou zde všechna čísla seřazena podle roku publikování. V každém roce se nachází adresáře s jednotlivými vydáními a v nich složky, se samotnými články a dalšími pomocnými soubory. V této kapitole bude popsána struktura jednoho vydání. Ostatní vydání se budou lišit pouze v počtu publikovaných článků a informacemi vstupujících do procesu tvorby.

Jak již bylo zmíněno, složka vydání (povětšinou reprezentovaná určitým číslem) obsahuje jednotlivé složky s články. Je potřeba dosáhnout toho, aby se tyto složky jmenovaly stejně, jako \LaTeX ový článek uvnitř těchto adresářů.

Kromě samotného článku ve formátu \LaTeX musí složka obsahovat všechny potřebné dokumenty, aby došlo ke správnému přeložení a převedení dokumentu do PDF. Adresář musí obsahovat i soubor `headfoot.tex`, který definuje vizuální podobu záhlaví a zápatí článku, jenž se přidává jednotlivým listům. *Tento soubor je taktéž potřeba brát jako soubor vzorový! A pro správnou funkci systému by se neměl měnit.* Pokud článek obsahuje obrázky, jsou většinou odděleny do speciálního podadresáře. Na obrázku č. 9 můžeme vidět obsah složky článku.

¹¹ <http://www.ctan.org/pkg/zwpagelayout>



Obrázek 9: Adresářová struktura článku

Kromě samotných článků musí adresář vydání obsahovat další důležité soubory, bez kterých by složení časopisu nebylo možné. Proto, pokud nebude nějaký z dále zmiňovaných souborů nalezen, bude proces tvorby ukončen.

7.5.3.1 about.txt Textový dokument obsahuje texty k vytvoření úvodního slova. V tomto souboru se musí dodržovat jistá pravidla, aby výsledné formátování bylo zobrazeno správně. Text obsažený v dokumentu je rozdělen na 3 části. Jednotlivé části jsou odděleny pomocí speciálních znaků -----. Každá část má svou vlastní podstatu. První sekci se definuje nadpis. V druhé části se objevuje samotný text a ve třetí je název autora, který se zobrazí pod jeho fotkou. Tento dokument je pevně svázán s souborem `Deska_V_vzor`, jenž byl popsán v kapitole 7.5.2.4 .

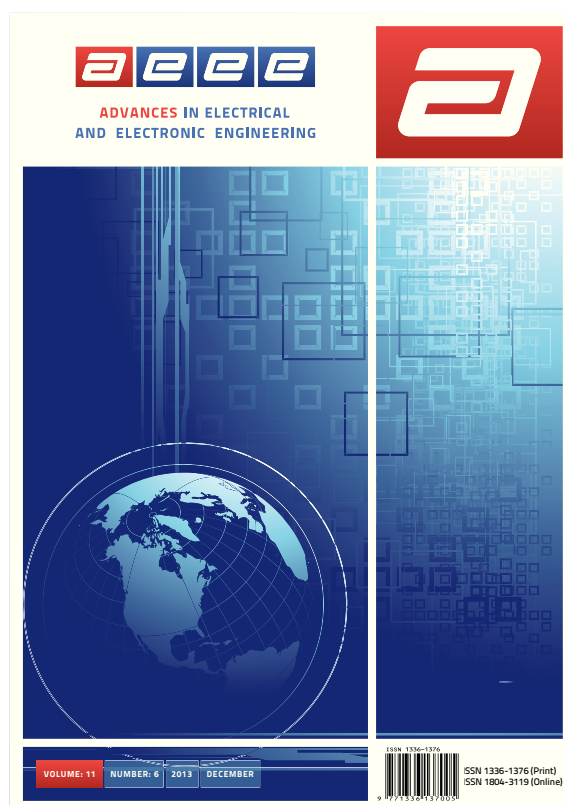
V samotném dokumentu, lze jednoduše tvořit jednoduchý formát výsledného textu pomocí klávesy ENTER. Podle toho kde, a kolikrát odřádkujeme, se tyto změny promítnou i do samotného výstupu v LaTeXu. Na obrázku č.10 lze pozorovat rozdíl mezi vstupním TXT souborem (na obrázku 10 v pravo), kde se pomocí odřádkování vytvořili odstavce a následný výstup převedený do PDF(na obrázku 10 v levo).

7.5.3.2 titulni_strana.pdf Tato stránka v PDF reprezentuje úvodní stranu obalu časopisu. Jelikož je to především grafická stránka, nebyla převáděna do \LaTeX ové podoby a je při překladu pouze importována. Redakcí je tato stránka upravována pomocí softwaru, který umožňuje otevřít formát AI. AI je formát používaný například programy jako Adobe Illustrator¹². Který umožňuje uložit projekty i do formátu PDF. Jeho výstup je můžeme vidět na obrázku číslo 11.

¹²Adobe Illustrator je komerční vektorový editor

<p>Dear readers,</p> <p>I have been associated with the technology of specialty optical material characterization since 2002. During this span of work, a solution doping technique through precise optimization of individual rare-earth doped optical fiber, fabrication of multi mode amplification hard fiber for UV-Vis-NIR zone etc. Recently I am working in the potential application in data communication and CO2 gas sensing.</p> <p>In my opinion, focus of present telecommunication research fibers targeting the fears of future "capacity-crunch" by implementation of (MDM).</p> <p>Already some remarkable progress has been achieved in the design and fabrication of few mode fibers but further improvement enhancing internet speed by 100X. On the other hand, in terms of ready achieved multi-kilowatt power level, further power enhancement avoid "photodarkening" phenomena that result in increased attenuation in the Yb-bands (915 - 976 nm) or even in the visible wavelength. Finally</p>	<p>Dear readers,</p> <p>I have been associated with the technology of special fabrication and their optical material characterization since 2002. During this span of work, enhancement of repeatability of solution doping technique through precise optimization of individual identification of new host glass for rare-earth doped optical fiber, fabrication of multi mode amplification pure silica core radiation hard fiber for UV-Vis-NIR zone etc. Recently I am working in two very initial results exhibit potential application in data communication and CO2 gas sensing.</p> <p>In my opinion, focus of present telecommunication research development of multi-mode specialty fibers targeting the fears of future "capacity-crunch" principle of Mode Division Multiplexing (MDM).</p>
--	---

Obrázek 10: Formátování textu PDF/TXT



Obrázek 11: Titulní stránka časopisu AEEE

7.5.3.3 foto.jpg Jedním ze souborů vstupujícím do celého procesu tvorby, zvláště pak do procesu vytváření úvodního slova je `foto.jpg`. Obrázek ve formátu JPG obsahuje fotku daného autora píšící úvodní slovo k časopisu. Jeho velikost se poté sama upraví v LaTeX ovém dokumentu `Deska_V.tex`.

7.5.4 Řídící soubor

Asi jeden u nejdůležitějších souborů v celém systému je soubor `source.csv`. Nachází se v každém vydání a definuje informace, podle kterých se vytváří záhlaví, zápatí, a obsah časopisu daného vydání. Časopis bude složen z těch částí a informací, které budou uvedeny v tomto dokumentu. Články, které se zde nenachází, nebudou připojeny do celkového dokumentu. Pokud by při kontrole vytvořeného časopisu byla nalezena chyba, je užitečné zkontrolovat právě tento řídicí soubor, zda se z něm nevyskytl překlep. CSV je formát, kde jednotlivé hodnoty jsou oddělené speciálními znaky. Může to být čárka, středník nebo tabulátor. V našem případě se v dokumentu pro oddělení jedlových údajů používá čárka. Proto je třeba mít na paměti, že při psaní hodnot, které obsahují čárky je potřeba, aby tyto položky byly psány v uvozovkách. Jedná se například o seznam autorů. Na obrázku č. 12 lze vidět názvy jednotlivých sloupců, které CSV soubor obsahuje.

	A	B	C	D	E	F	G	H	I
1	id	name	header	volume	number	start_page	year	month	authors
2	799-4366-1-LE	Fault Indic	Electrical and Electronic Engine	11	4	56	2014	FEBRUARY	Jan Moravec, Petr Novák
3	910-4716-1-LE	Statistical	Electrical and Electronic Engine	11	4	61	2014	FEBRUARY	Jakub Fidler, Jiří Mops
4									
5									

Obrázek 12: SOURCE.CSV

- `Id` udává jednoznačný název článku, kterým musí být pojmenovaný článek napsaný v \LaTeX u i samotná složka.
- `Name` název článku, tato kolonka současně slouží při vytváření obsahu.
- `Header` název určuje kategorii článku, jenž je obsahem záhlaví stránky.
- `Volume` je další hodnota uvedená v záhlaví.
- `Number` označuje číslo vydání.
- `Start_page` určuje, na které stránce bude daný článek začínat a dále je tato hodnota využívána u tvorby obsahu časopisu.
- `Year` určuje rok vydání.
- `Month` určuje měsíc vydání.
- `Authors` obsahuje seznam autorů, kteří článek psali, tyto informace jsou využívány při tvorbě obsahu časopisu.

Je důležité dát si pozor na správné názvy a velikosti písmen. Tento navržený systém poznává rozdíl mezi velkými a malými písmeny v jednotlivých řetězcích.

7.6 Skripty

Základními kameny celého systému jsou skripty, pomocí kterých se celý časopis sestavuje. Skripty jsou psány pomocí Bash což je unixový shell, který ztvárňuje příkazový řádek. Proces tvorby se spouští pomocí jediného skriptu *start.sh*, který se nachází v kořenovém adresáři. Tento skript poté volá ostatní skripty uložené v složce SKRIPTY. Následující podkapitoly podrobně popisují, co jednotlivé skripty provádí.

7.6.1 Start.sh

Tento skript slouží jako kořen, z kterého jsou volány další skripty jako jeho listy. Dalo by se říct, že na tento skript se napojují jednotlivé moduly, které provádí různé funkce během celé tvorby. V první fázi tvorby celého časopisu se musíme dostat do složky vydání, které chceme sestavit. Při tom se nás po spuštění program zeptá na rok a vydání, do kterého chceme přejít. (Obrázek č. 13).

```
student@student-virtual-machine:/home$ cd student/Desktop/CS1.02/
student@student-virtual-machine:~/Desktop/CS1.02$ sh start.sh
-----
Tvorba časopisu pro ukončení použijte  ctrl c
-----
2013  2014
-----
Zvolte rok:
2013
-----
10   11
-----
Zvolte vydání:
10
```

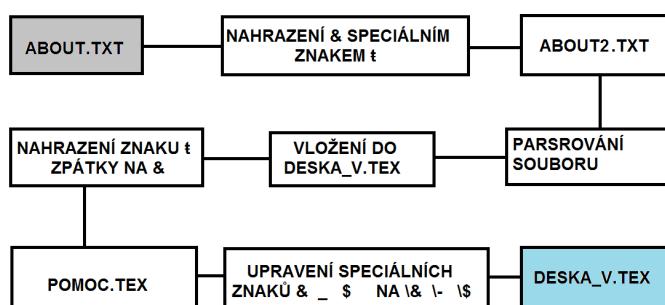
Obrázek 13: Volba vydání

Při začátku sestavování program klade otázky a vždy s otázkou se vypíší možnosti, které máme na výběr, tedy seznam roků, popřípadě vydání v daných letech. Pokud jsou zadávané informace chybné, systém na to upozorní a umožní nové zadání. V další části skriptu se ověří, zda existují pomocné soubory jako *about.txt*, *foto.jpg*, *source.csv* a *titulni_strana.pdf*. Všechny tyto soubory jsou podrobně popsány v kapitolách 7.5.3 a 7.5.4. Jestliže některý z uvedených souborů bude chybět, program bude s varovným hlášením ukončen. Následně jsou soubory zkopírovány na určená místa, kde budou čekat na zpracování. Základní skript se vždy nachází v úrovni kořenové složky. Cesta k zvolenému vydání je uložena do proměnné, která se jako parametr přidává při volání ostatních skriptů. Vývojový diagram toho skriptu je znázorněn na obrázku č. 19, jenž se nachází v příloze této práce.

7.6.2 About.sh

První skript, který je volán z hlavního skriptu. Má za úkol předpřipravit úvodní slova na přední straně časopisu. Respektive převést text ze souboru *about.txt* do \LaTeX ového souboru.

Jednoduché schéma toho procesu je znázorněno na obrázku č. 14. Celý postup tvorby začíná nahrazením speciálního znaku `&`. Je to především z důvodu, že `&` má v \LaTeX ových souborech zvláštní funkci 3.4.2 a tudíž tento znak není brán jako klasický text. Poté pokračuje procházením textového dokumentu, který je pomocí speciálních znaků rozdělen na 3 sekce (více v kapitole 7.5.3), vytvořený parser¹³, rozdělí jednotlivé sekce do tří proměnných. V dalším kroku tvorby proběhne přesunutí pomocného souboru `Deska_V_vzor.tex` a vytvoření dočasných složek. Poté se pomocí příkazu `sed` nahradí klíčová slova v `Deska_V_vzor.tex` (podrobnější informace v kapitole 7.5.2.4) a příkazem `TR` se nahradí zástupné znaky zpátky na `&`. Poslední funkcí tohoto skriptu, aby bylo docíleno správné zobrazení, je přidání lomítka k speciálním znakům v \LaTeX u. Podrobnější vývojový diagram se nalézá v příloze s obrázkem č. 20.



Obrázek 14: Proces nahrazování znaků

7.6.3 Clear.sh

Aby bylo možné skripty spouštět i několikrát po sobě, je musí se pomocné soubory a složky pročistit. K tomuto účelu se používá skript `clear.sh`. Který nejen čistí a vytváří pomocné složky, ale také seřadí řídicí soubor `source.csv`. Seřazení se provádí za pomoci příkazu znázorněném níže.

```
head -1 $INPUT > zdroj.csv | tail -n+2 $INPUT | sort -t , -k6
>> zdroj.csv
```

Nejprve skript za pomoci příkazu `head` zapíše do nového souboru `zdroj.csv` první řádek `source.csv` souboru, který obsahuje názvy jednotlivých sloupců. Je to z toho důvodu, aby se nám tyto hodnoty nepletly do třídění. `tail` vypíše řádky, které se mají setřídít (Tedy od druhého řádku níž). Tyto vybrané data poté příkaz `sort` setřídí vzestupně za pomoci šestého sloupce. Na této pozici se nachází sloupec počáteční strana, tímto se

¹³Cizí slovo vyjadřující program nebo funkci upravující textové řetězce

docílí správné pořadí článků. A celé se to uloží do již vytvořeného souboru `zdroj.csv`, se kterým se pracuje po zbytek průběhu tvorby časopisu.

Dalšími funkcemi toho skriptu jsou následné testování, jestli existují pomocné složky jako `celek`, `online`, `tisk`, `temp`. Pokud ne jsou vytvořeny. Tyto složky jsou dočasné a slouží pro shromažďování již vytvořených PDF dokumentů, čekající na další procesy. Celý proces lze pozorovat na vývojovém diagramu na obrázku č. 21 uvedeného v přílohách.

7.6.4 Uvod_preklad.sh

Za pomoci tohoto skriptu dochází k vytvoření úvodních listů časopisu. Je zde upravováno záhlaví a zápatí jednotlivých stran. Některé stránky již byly předpřipraveny jinými skripty jako například skriptem `about.sh`. Prvně jsou data načteny ze zdrojového `source.csv` do jednotlivých proměnných. Poté se zálohuje soubor `headfoot.tex`, který představuje záhlaví a zápatí stránek. Zálohuje se tím způsobem, že je soubor dočasně zkopírován do vytvořené složky `tmp`. Poté pomocí příkazu `SED` se nahradí staré údaje v záhlaví a zápatí údaji novými, které již obsahují správné proměnné. Jedná se o rok, měsíc, volume a vydání.

Takto připravené stránky jsou poté přeloženy upraveným příkazem, který skryje výpis překladu `pdflatex`.

```
pdflatex -interaction=batchmode "$s.tex" 1>/dev/null
```

Pokud překlad proběhne úspěšně je o tom vypsáno informativní hláška hlášení, v opačném případě se vypíše, že překlad selhal. Po úspěšném překladu se navrátí původní vzorový soubor, který je zálohovaný v `tmp` a smažou se soubory s příponou `.out`, `.log`, `.aux`, které byly vytvořeny při překladu `pdflatexem`.

Tento proces proběhne pro všechny úvodní stránky. Přesněji pro 8. Každé vytvořené PDF je následně nakopírováno do složky `CELEK`. Celý proces je znázorněn na vývojovém diagramu na obrázku č. 22 uvedený v příloze.

7.6.5 Uvod_spoj.sh

Po vytvoření všech úvodních stran časopisu, se musí tyto stránky pospojovat dohromady. Tuto funkci řeší skript `uvod_spoj.sh`. Jde o velice krátký a jednoduchý skript, který vstoupí do složky `celek` a pomocí příkazu `pdflatex` spojí jednotlivé strany do 2 souborů. Stránky úvodu 1 až 6 jsou sloučeny do souboru `front.pdf`. Druhý pdf dokument `back.pdf` obsahuje zbylé sloučené strany (ukázka níže). Po konečném sloučení pak bude mezi tyto dva dokumenty vložen PDF dokument obsahující jednotlivé články časopisu.

```
pdftk Deska_VII.pdf Deska_VIII.pdf cat output back.pdf
echo "Byl vytvořen konec"
```

7.6.6 Clanky_preklad.sh

Ve stejném duchu jako v kapitole 7.6.4 probíhá skript `clanky_preklad.sh`. Tento skript upravuje záhlaví a zápatí publikovaným článkům. Krom samotné úpravy skript vytvoří pomocný L^AT_EXový soubor, který je potom využit při tvorbě exportu metadat z daného článku. Skript začíná úplně stejně, načtením hodnot měnících se v záhlaví a zápatí článku. Jedná se o rok, měsíc, volume, vydání a kategorii článku. Podobně jak u překladu úvodu se zálohuje taky soubor `headfoot.tex`. Poté pomocí příkazu `SED` jsou staré údaje vyměněny za nové a pomocí příkazu `pdflatex` vytvořeny z L^AT_EXových souborů soubory PDF. Opět je příkaz upraven, aby se na obrazovku terminálu nevypisovaly rozsáhlé informace během překladu. Jak takový překlad probíhá lze vidět na obrázku č. 15, kde je zobrazen přehledný popis, jaké vstupní data jsou vkládána do článku ze zdrojového souboru CSV. A který článek se právě zpracovává.

```
ID : id
Name : name
Header : header
Volume : volume
Number : number
Start page : start_page
Year : year
Month : month
Autor : authors

ID : 799-4366-1-LE
Name : Fault Indicators of Partial Discharges in Medium-Voltage Systems
Header : Electrical and Electronic Engineering II
Volume : 11
Number : 4
Start page : 56
Year : 2014
Month : FEBRUARY
Autor : "Jan Moravec, Petr Novák"
```

Obrázek 15: Překlad článků

Po vytvoření všech článků jsou pdf soubory nakopírované do dočasné složky `TEMP`, ve které budou dále zpracovány. Samozřejmě při úspěšném překladu jsou opět smazány soubory s příponami `.out`, `.aux` a `.log`.

Dalším fází skriptu je vytvoření pomocného souboru pro tvorbu metadat. Tvořený pomocný soubor má strukturu nejjednoduššího L^AT_EXového dokumentu (viz kapitola 3.4.3). Pomocí příkazu `awk` s přepínačem `F`, který zaručuje nalezení prvního výskytu hledaného řetězce v článku s příponou `tex`. V tomto případě se jedná o řetězec `\Autor`, jenž je ukončený složenou závorkou a přidán do pomocného souboru.

```
awk -F: "/\Author{/,/}/{print;exit;}" "$idc/$idc.tex" >>pomoc.tex
```

Tyto příkazy se opakují pro všechny elementy, které jsou potřebné pro exportování metadat do XML. Takto přepracovaný soubor se pomocí programu `Tralics` převede na XML soubor. Ve skriptu je na to použit příkaz uveden níže.

```
tralics    pomoc.tex -conffdir="./" -silent -config=aeee.tcf -oe8
```

Přepínač `-conffdir` určuje, na kterém místě se nachází konfigurační soubor `aeee.tcf` popsán v kapitole 7.5.2.1. Přepínač `-slinet` utiší výpis logu při překladu, aby nedocházelo ke zbytečně dlouhým výpisům a tím k znehlednění celé tvorby a `-oe8` označuje kódování.

Výsledný soubor XML je poté přemístěno do složky `TEMP`, kde stejně jako samotné články čeká na další zpracování.

Celý tento proces probíhá v cyklu pro jednotlivé záznamy v `source.csv`. První cyklus je vynechán, protože jsou načítány hlavičky jednotlivých sloupců. Pokud by v tomto řídicím systému chyběl nějaký záznam o článku, který by měl patřit do časopisu nebude vytvořen. Vývojový diagram celého skriptu lze vidět na obrázku č. 23 uvedeného v přílohách.

7.6.7 Clanky_spoj.sh

Velice jednoduchý skript sloužící k spojení jednotlivých článků dohromady. Skript se opět řídí údaji uvedenými v `source.csv`, kdy se počítá s tím, že záznamy v CSV jsou seřazeny podle startovních stránek viz skript 7.6.3

```
[ ! -f $INPUT ] && { echo "$INPUT file not found"; exit 99; }
while read idc name header volume number start_page year
           month autor
do
    if [ -f "$idc.pdf" ]; then
        echo "Slucuji $idc.pdf"
        eval "pdftk $idc.pdf cat output claneke.pdf"
    fi
done < $INPUT
```

Výše je zmíněná ukázka kódu, jenž vytváří příkaz pro `pdftk`. `Pdftk` je program používající se pro práci s pdf v linuxu. Příkaz `eval` poté vytvářený řetězec spouští. A postupně vytváří konečný dokument `claneke.pdf`.

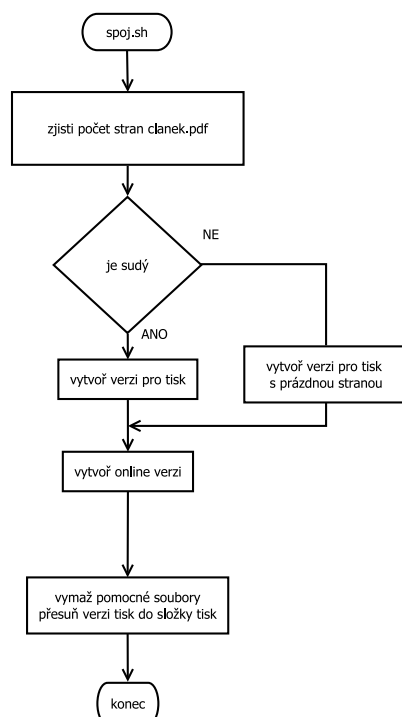
7.6.8 Spoj.sh

`Spoj.sh` je finálním skriptem, který jednotlivé předpřipravené části v dočasných složkách spojí dohromady. Tento proces probíhá ve složce `Online`. Aby skript mohl najít všechny umístění složky, jsou jednotlivé cesty připojeny při volání tohoto skriptu ze základního skriptu `start.sh`. Program nakopíruje všechny dokumenty z pomocných složek ve formátu PDF do adresáře `online`. Zde sestaví verzi `online` a verzi pro tisk. Verze `online` spojí dohromady tři dokumenty `front.pdf`, `clanky.pdf` a `back.pdf`.

Při tvorbě verze pro tisk si skript prvně ověří počet stránek souboru `clanky.pdf`. Za pomoci daného příkazu:

```
pocet=$(pdftk clank.pdf dump_data|grep NumberOfPages|awk '{print $2}')
```

Pokud je tento počet lichý, je přidána prázdná stránka. Poté je článek pro tisk přesunut do složky `Tisk` a nepotřebné dokumenty smazány. Jednoduchý diagram lze vidět na obrázku č. 16



Obrázek 16: Vyvojový diagram skriptu `spoj.sh`

7.6.9 Znacka.sh

Poslední skript, který se při celém procesu volá. Umožňuje přidat tiskové verzi časopisu ořezové značky. Skript využije pomocný soubor `casopis_tisk.tex`, který obsahuje nastavení pro ořezové značky. Na konci toho skriptu jsou vytvořeny 2 verze. Verze s ořezovými značkami obsahuje jednotlivé listy časopisu o velikosti A4, které umístěny na listy o velikosti A3 s přidánými značkami pro ořezání. Pokud by bylo potřeba parametry přenastavit, stačí upravit nastavení v již zmiňovaném pomocném souboru. Na obrázku č. 17 lze vidět list s ořezovými značkami.



Obrázek 17: List s ořezovými značkami

7.6.10 Souhrn

Po skončení průběhu tvorby systému, se vytvoří verze pro tisk a verze pro online publikování, které obsahují informace zadávané v konfiguračním souboru. V složce vydání se musí vždy nacházet potřebné doplňující dokumenty, potřebné ke korektnímu sestavení celého čísla. Během celého procesu, probíhá přesun pomocných souborů z určitých míst na jiné. Ve většině případů jsou tyto soubory umístěny ve složce *skripty* a přesouvány na potřebná místa.

Uživatel je během skládání časopisu informován na obrazovku terminálu, v které fázi se proces nachází a zda je úspěšný či nikoliv. Což umožňuje při výskytu případných chyb najít zdroj problémů či jejich řešení.

Navržená struktura spouštění skriptů zvyšuje přehlednost daného systému, umožňuje přidávat další skripty pro případ rozšíření této aplikace, popřípadě umožňuje odstranění určité části daného vytvořeného systému.

8 Testování vytvořeného systému

Další částí mé práce je otestování vytvořeného systému na libovolném open source redakčním systému určeného pro publikaci časopisů. Pro testování byl vybrán Open Jurnal System 6.1 (OJS 5), jenž jsem naistaloval na linuxovém systému Ubuntu 12.04.2. A ověřil jeho funkčnost.

8.1 První použití vytvořeného systému

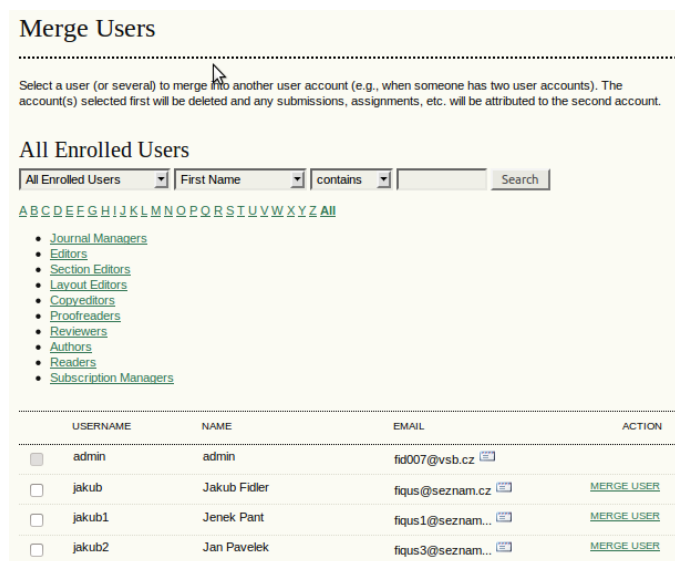
Pro samotné testování byl použit nově nainstalovaný operační systém. Takto připravený systém nemusí obsahovat všechny podpůrné programy, které jsou důležité pro správný chod celého systému. Proto je nutné je doinstalovat. Jednoduchý návod jak doinstalovat všechny potřebné programy se nachází v příloze B. Základním programem, používaným pro překlad \LaTeX ových souborů je Texlive. Nejvhodnější variantou je vždy instalovat nejaktuálnější verzi, protože u starších verzí můžou chybět některé balíčky, které jsou ve vytvořeném systému použity. Sám jsem použil plnou instalaci, abych se vyvaroval problémům s chybějícími balíčky. Dalším důležitým programem, bez kterého se neobejdeme je Tralics. Tento program, jak již bylo zmíněno, slouží pro překlad \LaTeX ových souborů na XML. A posledním softwarem, který je potřeba mít nainstalovaný je pdftk. Tenhle program slouží pro práci s PDF dokumenty. Ve vytvořeném systému se používá při slučování PDF do jednoho celku. Po doinstalování těchto programů, by měl systém korektně složit vydání časopisu.

8.2 Testování s OJS

Podle pokynů z kapitoly 6.2 jsem redakční systém naistaloval. Poté za pomoci administrátorského účtu byly vytvořeny dva rozdílné účty. Účet pro autora (osoba jenž vkládá články do systému 6.3.4) a druhý účet pro editora (osoba, která spravuje všechny články a řídí celé publikování 6.3.5) (Obrázek 18). Do systému bylo vloženo několik článků, které prošly jednotlivými fázemi publikování. Až se dostali do finální fáze tvorby dokumentu, kde bylo potřeba článkům přiřadit záhlaví a zápatí.

Jako editor, jsem si stáhl jednotlivé články z OJS do již připraveného adresáře pro budoucí vydání. Ověřil jsem si, zda se složky obsahující články se jmenují stejně jako samotné \LaTeX ové články. Z vyexportovaného csv souboru jsem si vytáhl jména autorů pro jednotlivé články, a upravil si řídicí soubor `source.csv` podle potřeby daného vydání.

Dále jsem do složky vydání přidal soubor `about.txt` s úvodním slovem, `foto.jpg` a upravenou přední stranu časopisu `predni_strana.pdf`. V terminálu jsem se pomocí příkazu `cd` dostal do kořenové složky vytvořeného systému. Přesněji do adresáře `CS1.02`, jenž se nachází na ploše. Poté pomocí příkazu `sh start.sh` jsem spustil celý proces. Program se mě prvně zeptal na číslo vydání, které chci vytvořit a poté už probíhalo samotné vytváření. Po velmi krátké době se mi v složce vydání objevili dvě složky s verzí pro tisk, které obsahuje dva pdf soubory jedno s ořezovými značkami a jedno bez. Druhá složka online obsahuje časopis pro online publikování a podsložku s jednotlivými



Obrázek 18: Ukázka uživatelů v OJS

časopisy a jejich vygenerovanými metadaty. Tyto PDF verze jsou posílány na revizi k poslední kontrole. Poté je již možno články publikovat. Pokud zde revizor našel nějaké nedostatky, nebyl problém chybu rychle opravit, většinou se chyba vyskytovala v souboru `source.csv`, kde díky přehlednosti byla rychle odhalena.

8.3 Zhodnocení testování

Hlavním výsledkem tohoto testování bylo ověření funkčnosti daného systému, který je schopný během velmi krátké chvíle vytvořit kompletní vydání časopisu. Tento systém bude využívat editor ve fázi těsně před publikací, kdy bude zapotřebí článkům přidat záhlaví a zápatí a všeobecně je upravit pro dané vydání. Celý systém je poté schopen každému definovanému vydání vytvořit pdf soubor s celým časopisem, které může editor poslat do tiskárny na tisk nebo publikovat na webových stránkách.

Důležité je dát si pozor na velikosti písmen a dodržování názvu vstupních souborů. Při záměně těchto názvů nedochází k vytvoření vydání. A uživatel je o tom informován chybovou hláškou.

Co se týče časové náročnosti, příprava vstupních souborů nezabrala ani desetinu času toho co ruční předělávání jednotlivých stránek časopisu. Zrychlení tvorby také napomáhá možnost exportování dat do formátu CSV, což bude editorovi ulehčovat práci při tvorbě řídicího `source.csv`. A zprovoznění systému na čisté instalaci Linuxu nezabírá také velké množství času, ale je důležité mít nainstalované všechny potřebné externí programy.

9 Závěr

Hlavním cílem této diplomové práce bylo vytvořit systém, který je schopný automatizovaně podle zadaných parametrů seskládat jednotlivé články časopisu do jednoho čísla. Při procesu vytvořit aktuální záhlaví a zápatí jednotlivých stránek. Ke článku poté vygenerovat metada v formátu XML za pomoci programu Tralics. Celý systém byl poté odzkoušen na čisté instalaci operačního systému Linux a vybraném redakčním systému s podporou publikování vědeckých časopisů.

Vytvořena práce je rozdělena na dvě základní části. První část je více zaměřená na teoretický popis technologií, které vstupují do vytvářeného systému. Obsahuje obecný popis vědeckého časopisu, pro který je tento systém sestaven. Dále je zde uveden základní popis \LaTeX technologie s přesnějším zaměřením na \LaTeX , u kterého je zmíněna tvorba základního dokumentu a výpis hrstky příkazů, jenž se nachází v odborných článcích psaných v \LaTeX u. Pro Tralics sloužící pro převod \LaTeX ových souborů do XML, používaný pro generování metadat, jsou zde uvedeny základní principy a konfigurace pro pochopení základních funkcí tohoto programu. Systém pracuje na principu bashovských skriptů, proto jsou zde uvedeny a teoreticky popsány základní příkazy pro práci se soubory. Aby bylo možno zjistit přesné umístění, kde tento vytvořený systém použít, jsou v této práci teoreticky rozebrány jednotlivé role, které se používají v procesech tvorby časopisu. Přesněji jsou to role definované Open Jurnal Systémem.

Druhá část se zabývá praktickým řešením vytvářeného systému. Jsou zde velice jednoduše popsány možnosti řešení. Výchozím řešením bylo zvoleno vytvořit systém pomocí bash skriptů, které se zdály pro méně zkušenější uživatele více přehlednější v případné budoucí úpravě systému podle požadavku, které vyplynou až z dlouhodobého testování vytvořeného systému.

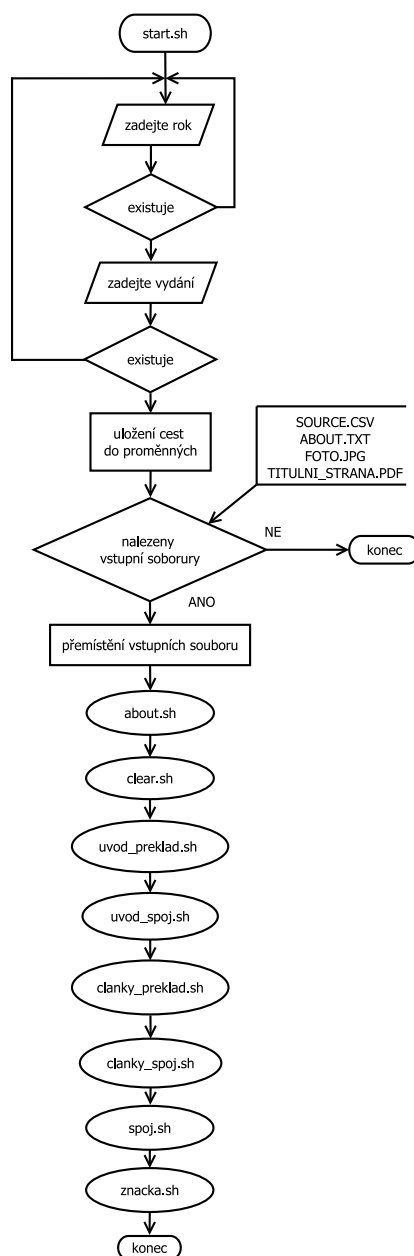
Poté již jsou popsány jednotlivé procesy, které probíhají při skládání daného vydání. Je zde popsána adresářová struktura. U jednotlivých adresářů poté vysvětleny a popsány jednotlivé funkce souborů, které obsahují. Podrobněji jsou popsány samotné skripty a pomocné soubory, které vstupují do procesu tvorby a významně ovlivňují výsledný vzhled skládaného časopisu. V této části se také nachází testování nejprve na čisté instalaci operačního systému Linux a poté i s redakčním systémem OJS. Při tomto testování byly dolazeny poslední nesrovnalosti. Nalezeny výhody OJS. Především možnost exportování vybraných dat do csv souboru, což umožňuje snadnější tvoření řídicího souboru. Otestováním jsem zjistil, že mnou vytvořený systém by měl používat Editor ve fázi, kdy jsou jednotlivé články schváleny k publikování a potřebují pouze přidat záhlaví a zápatí. V tomto procesu se také vytvoří celý samotný časopis, který je poté ještě přezkontrolován a poté publikován online na webu, popřípadě dán do tisku.

10 Reference

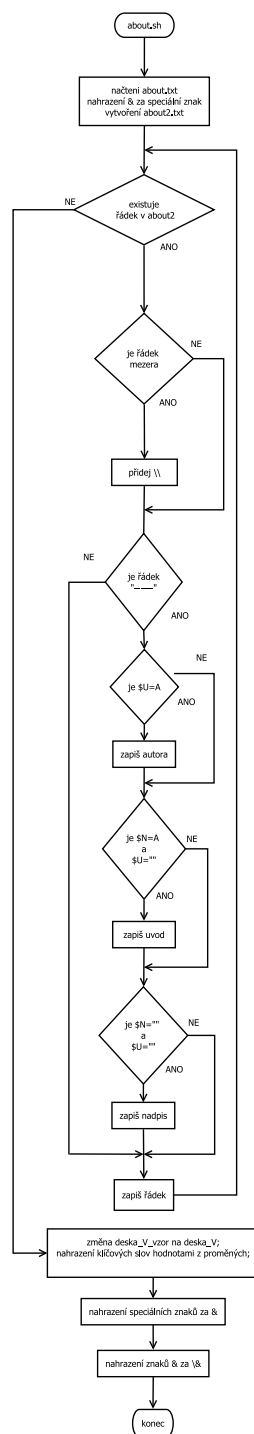
- [1] Kosek, Jiří *Co je to TeX?*. Kosek.cz. [online]. 1999 [cit. 2014-04-22]. Dostupné z: <http://www.kosek.cz/clanky/cw/cojetex.html>
- [2] Oetiker, T., a kol. *Ne příliš stručný úvod do systému L^AT_EX2_ε* 1998.
- [3] Beneš Miroslav *Úvod do systému LaTeX. Stránky pro vyuku*. [online]. 1998 [cit. 2014-04-22]. Dostupné z: <http://www.cs.vsb.cz/benes/vyuka/latex/uvod.htm>
- [4] *Journal History*. AEEE. [online]. 2014 [cit. 2014-04-22]. Dostupné z: <http://advances.utc.sk/index.php/AEEE/about/history>
- [5] *Editorial Policies*. AEEE. [online]. 2014 [cit. 2014-04-22]. Dostupné z: <http://advances.utc.sk/index.php/AEEE/about/editorialPolicies#peerReviewProcess>
- [6] Martinek David *BASH*. [online]. 2013 [cit. 2014-04-22]. Dostupné z: <http://www.fit.vutbr.cz/~martinek/gymnazium/bash.html.cs>
- [7] Watzke David *Unixové nástroje-7*. ABCLinuxu. [online]. 2010 [cit. 2014-04-22]. Dostupné z: <http://www.abclinuxu.cz/clanky/navody/unixove-nastroje-7-tr-cut-sort-a-uniq#sort>
- [8] Watzke David *Unixové nástroje-1*. ABCLinuxu. [online]. 2010 [cit. 2014-04-22]. Dostupné z: <http://www.abclinuxu.cz/clanky/navody/unixove-nastroje-1-uvod-cat-head-tail#tail>
- [9] Macek Petr *Hrátky z řádky*. Root.cz. [online]. 2008 [cit. 2014-04-22]. Dostupné z: <http://www.root.cz/clanky/hratky-z-radky-tr-sed-awk-a-ti-dalsi/>
- [10] *Programování v (bash) shellu*. Root.cz. [online]. 2000 [cit. 2014-04-22]. Dostupné z: <http://www.root.cz/clanky/programovani-v-bash-shellu/>
- [11] *Configuration files of Tralics*. Tralics. [online]. 2008 [cit. 2014-04-22]. Dostupné z: <http://www-sop.inria.fr/marelle/tralics/titlepage.html>
- [12] *Options of the Tralics program*. Tralics. [online]. 2013 [cit. 2014-04-22]. Dostupné z: <http://www-sop.inria.fr/marelle/tralics/options.html>
- [13] *Getting started with Tralics*. Tralics. [online]. 2008 [cit. 2014-04-22]. Dostupné z: <http://www-sop.inria.fr/marelle/tralics/doc-step.html>
- [14] *Budování a provoz digitální knihovny s přihlédnutím k bezpečnosti*. Brno, 2011. Diplomová práce. Masarykova Univerzita.
- [15] Rumánek Martin *Open Jurnal Systems*. [online]. 2012 [cit. 2014-04-22]. Dostupné z: <https://dspace.muni.cz/jspui/bitstream/ics.muni.cz/985/1/esej.pdf>
- [16] *Open Journal Systems. Public Knowledge Project*. [online]. 2008 [cit. 2014-04-22]. Dostupné z: <https://pkp.sfu.ca/ojs/>

-
- [17] *Open Journal Systems. Public Knowledge Project. [online].* 2008 [cit. 2014-04-22]. Dostupné z: <http://pkp.sfu.ca/ojs/ojs-usage/ojs-stats/>
- [18] *System Requirements. Public Knowledge Project. [online].* 2010 [cit. 2014-04-22]. Dostupné z: <http://pkp.sfu.ca/ojs/docs/userguide/2.3.3/overviewSystemRequirements.html>
- [19] *About User Roles. Public Knowledge Project. [online].* 2010 [cit. 2014-04-22]. Dostupné z: <http://pkp.sfu.ca/ojs/docs/userguide/2.3.3/basicsRolesOverview.html>
- [20] *Site Administrator. Public Knowledge Project. [online].* 2010 [cit. 2014-04-22]. Dostupné z: <http://pkp.sfu.ca/ojs/docs/userguide/2.3.3/siteAdministrationOverview.html>
- [21] *Journal Management. Public Knowledge Project. [online].* 2010 [cit. 2014-04-22]. Dostupné z: <http://pkp.sfu.ca/ojs/docs/userguide/2.3.3/journalManagementOverview.html>
- [22] *Authors. Public Knowledge Project. [online].* 2010 [cit. 2014-04-22]. Dostupné z: <http://pkp.sfu.ca/ojs/docs/userguide/2.3.3/authorOverview.html>
- [23] *Editors. Public Knowledge Project. [online].* 2010 [cit. 2014-04-22]. Dostupné z: <http://pkp.sfu.ca/ojs/docs/userguide/2.3.3/editorOverview.html>
- [24] *Section Editors. Public Knowledge Project. [online].* 2010 [cit. 2014-04-22]. Dostupné z: <http://pkp.sfu.ca/ojs/docs/userguide/2.3.3/sectionEditorOverview.html>
- [25] *Reviewers. Public Knowledge Project. [online].* 2010 [cit. 2014-04-22]. Dostupné z: <http://pkp.sfu.ca/ojs/docs/userguide/2.3.3/reviewerOverview.html>
- [26] *Copyeditors. Public Knowledge Project. [online].* 2010 [cit. 2014-04-22]. Dostupné z: <http://pkp.sfu.ca/ojs/docs/userguide/2.3.3/copyEditorOverview.html>
- [27] *Layout Editors. Public Knowledge Project. [online].* 2010 [cit. 2014-04-22]. Dostupné z: <http://pkp.sfu.ca/ojs/docs/userguide/2.3.3/layoutEditorOverview.html>
- [28] *Proofreaders. Public Knowledge Project. [online].* 2010 [cit. 2014-04-22]. Dostupné z: <http://pkp.sfu.ca/ojs/docs/userguide/2.3.3/layoutEditorOverview.html>
- [29] *Readers. Public Knowledge Project. [online].* 2010 [cit. 2014-04-22]. Dostupné z: <http://pkp.sfu.ca/ojs/docs/userguide/2.3.3/readerOverview.html>

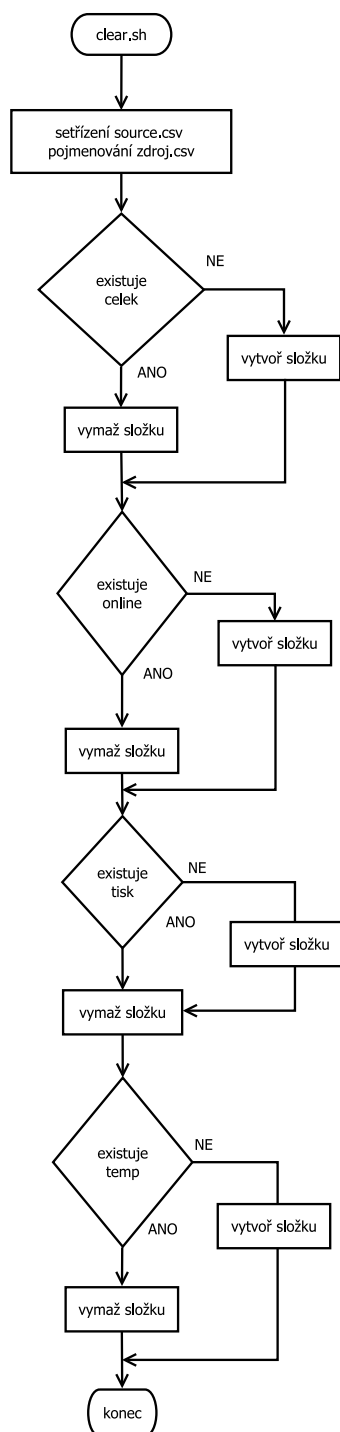
A Vývojové diagramy bash skriptů

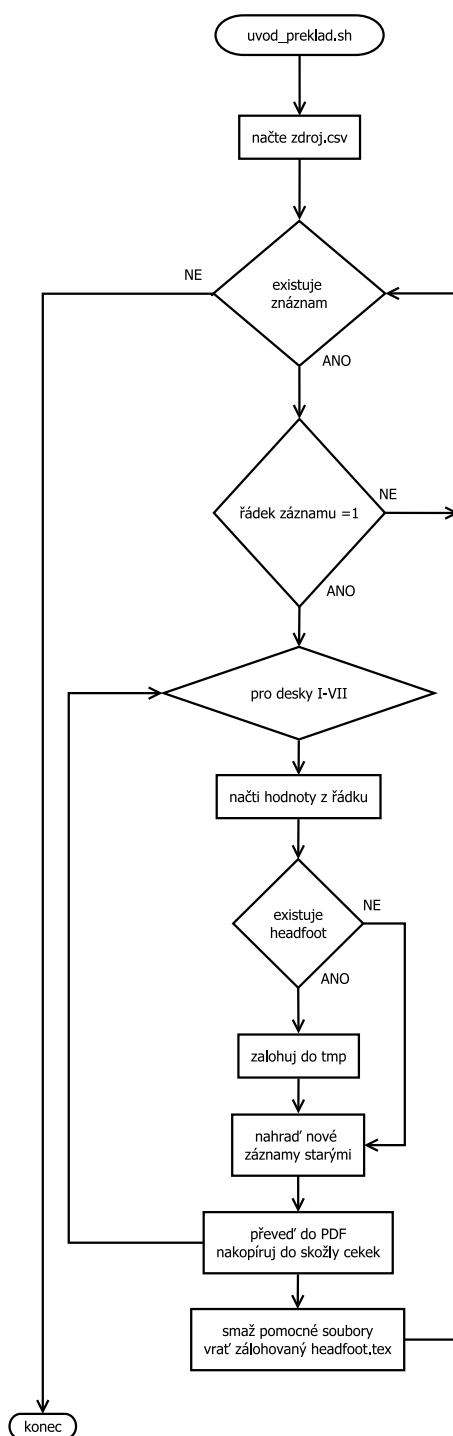


Obrázek 19: Vývojový diagram skriptu `start.sh`

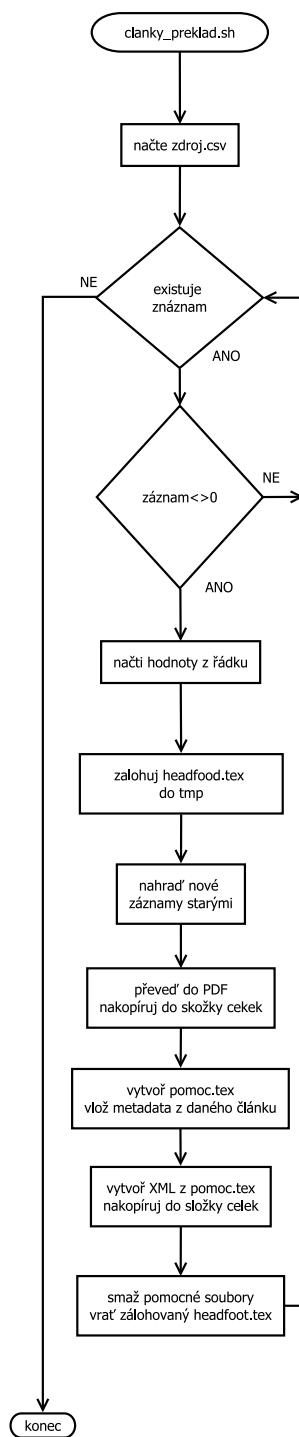


Obrázek 20: Vývojový diagram skriptu about.sh

Obrázek 21: Vývojový diagram skriptu `clear.sh`



Obrázek 22: Vývojový diagram skriptu `uvod_preklad.sh`



Obrázek 23: Vývojový diagram skriptu clanky_preklad.sh

B Postup zprovoznění systému na OS Linux

Celý navržený systém, se nachází na přiloženém DVD(Jeho obsah je v příloze C. Aby byla zaručena plná funkčnost, předpokládá se, že operační linuxový systém má už naistalované určité programy. Pokud tomu tak není, v této příloze je postup jak všechny potřebné programy doinstalovat. Jako čistý operační systém je zvolen LINUX Ubuntu 12.04.2.

První program pro správné fungování je TEXTLIVE 2012

Pomocí příkazu přidáme repozitář, pro možnost stáhnutí aktuální verze texlive.

```
sudo add-apt-repository ppa:texlive-backports/ppa
```

Poté provedeme update .

```
sudo apt-get update
```

A naistalujeme pomocí příkazu Texlive.

```
sudo apt-get install texlive-full
```

Druhým potřebným programem je Tralics.

Pomocí příkazu otevřeme v texovém editoru sources.list.

```
sudo gedit /etc/apt/sources.list
```

A ověříme zda se zde nachází daný záznam. Pokud ne dopíšeme jej.

```
deb http://us.archive.ubuntu.com/ubuntu raring main universe
```

Poté provedeme update.

```
sudo apt-get update
```

Naistalujeme Tralics.

```
sudo apt-get install tralics
```

Třetím programem je Pdftk.

Ten naistalujeme jednoduše.

```
sudo apt-get install pdftk
```

C Obsah DVD

Přiložené DVD obsahuje následující programy a dokumenty:

- DP.pdf - diplomová práce
- CS1.02.rar - obsahuje daný systém
- Ubuntu_CS.rar - obraz operačního systému Linux s vytvořeným systémem.
- hesla.txt - soubor s hesly k OS vytvořenému na virtuálnímu pc.
- VMware Player for Linux 32-bit -program pro spuštění na OS Linux
- VMware Player for Windows 32-bit and 64-bit - program pro spuštění na OS Win